With the support of the
Erasmus+ Programme
of the European Union

EMJD-SEED PROGRAMME

Proper Generalized Decomposition solutions for composite laminates with parametrized fibre orientations for fast computations

by

Karim El-Ghamrawy

A thesis submitted to

Dipartimento di Ingegneria Civile ed Architettura

Università di Pavia

and

Departament d'Enginyeria Civil i Ambiental

Universitat Politècnica de Catalunya

in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Computational Mechanics and Advanced Materials

Supervisors: Prof. Ferdinando Auricchio Prof. Pedro Díez

Co-supervisor and programme coordinator: Dr. Sergio Zlotnik

Pavia & Barcelona November 2020

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed:  Date: 18-11-2020

STATEMENT 1

This thesis is the result of my own investigations, except where otherwise stated. When correction services have been used, the extent and nature of the correction is clearly marked in a footnote(s). Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed:  Date: 18-11-2020

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed:  Date: 18-11-2020

To the souls of my father, my grandmother, and Nelly who I have lost during the PhD. You've always set an example for me in love, support, wisdom, and patience. Thank you for everything...

Acknowledgements

I would like to express my sincerest gratitude to my supervisors Prof. Ferdinando Auricchio, Prof. Pedro Díez, and Dr. Sergio Zlotnik for being always by my side on both the academic and personal level. This thesis would not be possible without their guidance and patience. A very special thanks to Dr. Gianluca Alaimo for his mentorship and great help. A very special thanks to the administration team: Lelia, Sonia, and Virginie.

I would like to extend my gratitude to the European Commission EACEA Agency for providing me the financial assistance under the Framework Partnership Agreement 2013-0043 Erasmus Mundus Action 1b as a part of the EM Joint Doctorate Simulation in Engineering and Entrepreneurship Development (SEED).

Throughout the PhD, I have met amazing people. I would like to thank the Pavia team: Alessandro, Stefania, Simone, Michele, Giulia, Cattenone, Margherita, Valentina, Alice, Xi, Massimo, Franca, Alessia, Laura, Rod, guillermo, Alex V., Anna, Gaetano, Varun, Kayal, Sai, and Michele T. I would also like to thank my Barcelona gang: Berto, Alex K., Arash, Hasini, Simona, David, La Spina, Paff, Fabi, Christina, Boyi, Ygee, and Jonatha.

I would also like to extend my gratitude to Osama, Gianmaria, Lorenzo, Alfredo, Murer, Ahmed for being more than just colleagues and truly great friends. I will always be in debt for Alberto Sibileau and Olga for supporting me while going together through thick and thin; and for the endless conversations trying to push each forward and fix the world.

A very warm and special thanks from my heart to Orsi, for always being there in every possible way.

Last but not least, the people who are behind everything, my family. I will always be in debt for my mom and my brother for everything they've done for me to make this PhD work and come this far. I also thank my friends in Egypt who were always there despite the distance.

Sommario

I materiali compositi stanno guadagnando popolarità come alternativa ai materiali classici in molte diverse applicazioni. Inoltre, il loro design è sempre più flessibile grazie al potenziale della manifattura additiva. Pertanto, è possibile produrre un laminato composito su misura con i valori ottimali di alcuni parametri di progettazione allo scopo di fornire le prestazioni meccaniche desiderate. In questo contesto, disporre di un modello numerico parametrico per la risposta meccanica del laminato composito è essenziale per calcolare i parametri ottimali. In generale, risolvere un modello meccanico utilizzando tecniche basate su mesh in 3D è computazionalmente costoso e ad un certo punto potrebbe diventare irrealizzabile quando il problema è multidimensionale. Inoltre, se il problema in esame è un'applicazione che richiede più *query* nei problemi di ottimizzazione, i problemi inversi o la quantificazione dell'incertezza, il problema diretto viene risolto numerose volte aumentando drasticamente il carico computazionale.

Nella presente tesi, i parametri di progetto presi in considerazione sono gli angoli che descrivono l'orientamento delle fibre di rinforzo nei diversi strati o nelle diverse zone dei laminati compositi. Presentiamo il criterio di rottura di Tsai-Wu come funzione obiettivo del problema di ottimizzazione. Nel presente lavoro, si raccomanda l'uso di una tecnica di "Model Order Reduction" (MOR) per decrementare il costo computazionale menzionato; ovvero, si ricorre alla "Proper Generalized Decomposition" (PGD) per ottenere la soluzione generalizzata della risposta meccanica della struttura. In particolare, otteniamo un *vademecum* computazionale 3D che fornisce un indice di rottura del laminato e un fattore di sicurezza che dipendono esplicitamente dall'orientamento della fibra. PGD fornisce anche sensitività per un algoritmo di ottimizzazione basato sul gradiente. La potenzialità e l'efficacia dell'approccio presentato è dimostrata attraverso alcuni test numerici. Infine, viene presentato un accoppiamento tra la metodologia proposta e le tecniche di *clustering* per migliorare le prestazioni complessive del modello.

Abstract

Composite materials are gaining popularity as an alternative to classical materials in many different applications. Moreover, their design is even more flexible due to the potential of additive manufacturing. Thus, one can produce a tailored composite laminate with the optimal values of some design parameters providing the desired mechanical performance. In this context, having a parametric numerical model for the mechanical response of the composite laminate is essential to compute the optimal parameters. Generally, solving a mechanical model using mesh-based techniques in 3D is computationally expensive and at some point it could become infeasible when the problem is multidimensional. Furthermore, if the problem under consideration is an application requiring multiple queries such as optimization, inverse problems, or uncertainty quantification, the direct problem is solved numerous times increasing drastically the computational burden.

In the present thesis, the design parameters under consideration are the angles describing the orientation of the reinforcement fibers in different layers or patches of the composite laminates. We present the Tsai-Wu failure criterion as the objective function of the optimization problem. The use of a Model Order Reduction (MOR) technique is advocated to alleviate the mentioned computational burden. Namely, we resort to the Proper Generalized Decomposition (PGD) to obtain the generalized solution of the structure mechanical response. Particularly, we obtain a computational vademecum which provides laminate failure index and safety factor that depend explicitly on the fiber orientation. The PGD vademecum provides also sensitivities for a gradient-based optimization algorithm. The potentiality and efficiency of the presented approach is demonstrated through some numerical tests. Finally, a coupling between the proposed methodology and clustering techniques is presented to enhance the overall performance of the model.

Keywords: Generalized solutions; Composite laminates; Fibre orientation; Failure; Optimization; PGD; Clustering optimization; 3D printing

"Strive not to be a success, but rather to be of value."

Albert Einstein

CONTENTS

Title **Page No.**

Sommario ix

Abstract x

List of Figures xvi

List of Tables xxi

List of Algorithms xxii

Abbreviations xxii

Nomenclature xxv

1 Introduction 1

1.1 Motivation . . . 3

1.2 Background . . . 4

1.3 State of the art . . . 9

1.3.1 Optimization of fibre orientation in composite laminates . . . 9

1.3.2 PGD as a MOR technique and its applications . . . 11

1.4 Scope and outline of this thesis . . . 12

2 Problem statement 14

2.1 Preliminaries . . . 14

2.2 Governing equations and constitutive law . . . 15

2.3 Material parametrization . . . 19

2.4 Failure criteria . . . 23
2.4.1 Maximum stress, maximum strain, and Tsai-Hill criteria . . . 24
2.4.2 Tsai-Wu criterion . . . 25
2.5 Optimization problem . . . 29

3 Proper Generalized Decomposition (PGD) framework 33
3.1 Preliminaries . . . 33
3.2 PGD brief introduction . . . 36
3.3 PGD formulation at a glance . . . 37
3.4 Encapsulated PGD . . . 40
3.5 Separation of input for PGD solver . . . 42
3.6 Post-process and sensitivities . . . 43

4 Numerical examples 49
4.1 Preliminaries . . . 49
4.2 Plate under tensile load . . . 53
4.3 Plate with circular hole under tensile load . . . 62

5 Domain decomposition using data analysis techniques for efficient PGD parameterization 73
5.1 Preliminaries . . . 73
5.2 Clustering optimization for PGD: motivation and objectives . . . 75
5.2.1 Preanalyses . . . 77
5.2.2 Principal Component Analysis . . . 78
5.2.3 Clustering of factors and their intersection . . . 84
5.2.4 Error computation and clustering optimization . . . 89
5.2.5 Fibre orientation optimal results . . . 93
5.3 Introduction to Additive Manufacturing (AM) and experimental validation 97

6 Concluding remarks 105
6.1 Summary . . . 105
6.2 Concluding remarks . . . 105
6.3 Future work . . . 107

A Collection of terms of the elasticity tensor **109**

B Separation terms of the elasticity and transformation tensors **111**

C Material characteristics of carbon fibre reinforced ABS **116**

D Clustering techniques **118**

D.1 K-means 118

D.2 Hierarchical clustering 124

E Mathematics of Principal Components **128**

F Factors of the raw safety factor data and its clustering **130**

Bibliography **134**

LIST OF FIGURES

Title **Page No.**

1.1 Structure of Chapter 1 2

1.2 Different composite materials in nature 5

1.3 An example of a complex geometry possible with 3D printing 6

1.4 Four levels of structural optimization. Figure adapted from [1] 7

2.1 Structure of Chapter 2 15

2.2 Independent directions in orthotropic material. Figure from [2] with permission. 20

2.3 Transversely isotropic material with yz-plane as the plane of isotropy . . 20

2.4 Global coordinate system {O,x,y,z} and material coordinate system {O,1,2,3} 21

2.5 Convex and non-convex functions example 30

3.1 Structure of Chapter 3 35

4.1 Structure of Chapter 4 51

4.2 Simulation workflow 52

4.3 Plate under tensile load of 45° 53

4.4 Convergence curve and amplitude evolution for plate under tensile load of 45° with greedy tolerance $\xi = 10^{-4}$ 54

4.5 Convergence curves for plate under tensile load of 45° with different greedy algorithm tolerances 55

4.6 Amplitude evolution of uncompressed and compressed PGD solution for plate under tensile 45°for various greedy algorithm tolerances ξ 56

4.7 Comparison of amplitudes of PGD compressed solution between fine and coarse spatial meshes 56

4.8 Displacement magnitude for $\theta_1 = -45°$ and $\theta_2 = 0°$ 57

4.9 Normalized spatial modes for $\theta_1 = -45°$ and $\theta_2 = 0°$ 58

4.10 PGD compressed solution: Normalized parametric functions 60

4.11 Failure criteria maps of plate under tensile load at 45°. 60

4.12 Failure criteria maps of plate under small tensile load at 45°with reduced applied load. 61

4.13 Failure criteria maps of plate under tensile load at: (a) & (b) 30°, (c) & (d) 60°. 62

4.14 Symmetric half of a square plate with a circular hole: 2 subdomains . . . 63

4.15 Plate with hole under tensile load at 0° with two parameters for a greedy tolerance $\xi = 10^{-4}$ 63

4.16 Plate with hole under tensile load at 0° with two parameters for a greedy tolerance $\xi = 10^{-6}$ 64

4.17 Plate with hole under tensile load at 0° with two parameters: Objective functions maps 65

4.18 Symmetric half of a square plate with a circular hole 65

4.19 Amplitude of the PGD modes and the error in space measured in one parametric point 66

4.20 Optimal fibre orientations on the deformed domain obtained by a global maximization of the safety factor λ_s using first the Genetic Algorithm and then a gradient-based method. 67

4.21 Symmetric half of a square plate with a circular hole subdivided in 8 patches 69

4.22 Amplitude of the PGD compressed solution and the error in space measured in one parametric point, $\boldsymbol{\theta}^{\mathtt{Opt}}$, defined inTable 4.4 70

4.23 Optimal fibre orientation on deformed domain 70

4.24 CPU time evolution of PGD vs standard FEM 72

5.1 Structure of Chapter 5 75

5.2 Example of two partitioned domain with four sub-divisions 76

5.3 Snapshots of different fibre orientations in the domain 77

5.4 Raw data distribution histograms 78

5.5 Histogram of the transformed safety factor raw data $\log(\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}})$ 79

5.6 PCA example between two features 80

5.7 Evolution of the cumulative variance with the number of factors 81

5.8 The first four factors obtained from applying PCA on stress raw data . . 82

5.9 The first four factors obtained from applying PCA on the transformed safety factor raw data matrix $\log(\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}})$ 83

5.10 Clustering of the first four factors obtained from applying PCA on the stresses raw data matrix using K-means. Each factor is divided into 2,3,4 clusters from left to right. 85

5.11 Clustering of the first four factors obtained from applying PCA on the stresses raw data matrix using Ward's method. Each factor is divided into 2,3,4 clusters from left to right. 86

5.12 Clustering of the first four factors obtained from applying PCA on the transformed safety factor raw data matrix using K-means. Each factor is divided into 2,3,4 clusters from left to right. 87

5.13 Clustering of the first four factors obtained from applying PCA on the transformed safety factor raw data matrix using Ward's method. Each factor is divided into 2,3,4 clusters from left to right. 88

5.14 Partition with 8 clusters resulting from the intersection of 4 factors where each of them was clustered using Ward's method. The number of clusters for factors 1,2,3,4 is 1,2,1,4 respectively. 89

5.15 Clustering optimization solutions obtained from Algorithm 5.1 using K-means with a single run and 10 runs. 92

5.16 Clustering optimization solutions obtained from Algorithm 5.1 using Ward's method 93

5.17 Domain parameterization for PGD solver based on the clustering optimization of the stress data using: K-means in (a) & (b) and Ward's method in (c) & (d) 94

5.18 Domain parameterization for PGD solver based on the clustering optimization of the transformed safety factor data using: K-means in (a) & (b) and Ward's method in (c) & (d) 95

5.19 Example of optimal fibre orientation for domains with 4 and 8 parameters obtained using the clustering of the safety factor via K-means 96

5.20 Weighting functions example. Illustration adapted from [3] 97

5.21 The Additive Manufacturing (AM) process. 98

5.22 FDM technology system scheme 98

5.23 Common infill patterns found in commercial software 99

5.24 Example of a cube with custom infill pattern generated by modifying the slicing software CuraEngine 100

5.25 Optimal fibre orientation for domains based on symmetrical partitioning and on clustering results using ABS material 101

5.26 Component designed with grips for the tensile test 102

5.27 Optimal fibre orientation for domains based on symmetrical partitioning and on clustering results using ABS material 102

5.28 Load vs extension curves for specimens with symmetrical and clustered based partitioning 103

5.29 Failure of structure for domains based on symmetrical partitioning and on clustering resulting from tensile tests 104

D.1 Example data set 119

D.2 Initial and final centroids positions and cluster solution for an example with $K = 2$ 120

D.3 Steps for random initialization of cluster centroids for an example with $K = 2$ 121

D.4 Bad initialization of cluster centroids for an example with $K = 2$ 122

D.5 Local minimum solution from the minimization of the K-means objective function 123

D.6 Cluster tree or dendrogram example 124

D.7 Types of linkage distances in hierarchical clustering 126

F.1 Evolution of the cumulative variance with the number of factors for the raw safety factor data 130

F.2 The first four factors obtained from applying PCA on the safety factor raw data matrix $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}}$ 131

F.3 Clustering of the first four factors obtained from applying PCA on the safety factor raw data matrix using K-means. Each factor is divided into 2,3,4 clusters from left to right. 132

F.4 Clustering of the first four factors obtained from applying PCA on the safety factor raw data matrix using Ward's method. Each factor is divided into 2,3,4 clusters from left to right. 133

LIST OF TABLES

Title **Page No.**

4.1 Optimized angles for square plate with circular hole for different objective functions. 67

4.2 Optimized angles for square plate with circular hole with different applied load values. 68

4.3 Different GA precision yielding values used as initial guesses for the gradient-based method 68

4.4 Different number of evaluations yielding different GA precision 71

5.1 Pareto set error comparison between K-means with a single run, K-means with 10 runs, and Ward's method 92

5.2 Safety factor index λ_s obtained from PGD based on different domain parameterization shown in Fig. 5.17 and Fig. 5.18 94

C.1 Carbon fibre reinforced ABS material characteristics 117

LIST OF ALGORITHMS

Title **Page No.**

2.1 Pseudo-code of the genetic algorithm 32

3.1 Pseudo-code of the FE loop to build the separated stiffness matrix for the PGD routines 43

5.1 Pseudo-code for clustering optimization algorithm 91

ABBREVIATIONS

PGD	**P**roper **G**eneralized **D**ecomposition
MOR	**M**odel **O**rder **R**eduction
PMC	**P**olymer **M**atrix **C**omposites
MMC	**M**etal **M**atrix **C**omposites
CMC	**C**eramic **M**atrix **C**omposites
CaMC	**Ca**rbon **M**atrix **C**omposites
RTM	**R**esin **T**ransfer **M**olding
AM	**A**dditive **M**anufacturing
FEM	**F**inite **E**lement **M**ethod
PDE	**P**artial **D**ifferential **E**quation
DM	**D**ata **M**ining
CM	**C**omputational **M**echanics
GA	**G**enetic **A**lgorithm
RB	**R**educed **B**asis
POD	**P**roper **O**rthogonal **D**ecomposition
PCA	**P**rincipal **C**omponent **Analysis**
SVD	**S**ingular **V**alue **D**ecomposition
BVP	**B**oundary **V**alue **P**roblem
FE	**F**inite **E**lements
SA	**S**imulated **A**nnealing
ABS	**A**crylonitrile **B**utadiene **S**tyrene

CPU	**C**entral **P**rocessing **U**nit
SSE	**S**um **S**quare **E**rror
CAD	**C**omputer **A**ided **D**esign
STL	**ST**ereo**L**ithography
FDM	**F**used **D**eposition **M**odelling
SLS	**S**elective **L**aser **S**intering
LOM	**L**aminated **O**bject **M**anufacturing

NOMENCLATURE

Scalars

$\mathtt{n_d}$	Number of global degrees of freedom
$\mathtt{n_{ne}}$	Number of nodes per element
$\mathtt{n_{nt}}$	Total number of nodes in the domain
$\mathtt{n_{edof}}$	Number of element degrees of freedom
N^e	Finite element shape function
Ω^e	Element domain
Ω	Physical domain
θ	Orientation of the fibres w.r.t the global coordinates
$\mathtt{n_p}$	Number of parameters
$\sigma_{\mathtt{y}}^{Lc}$	Longitudinal compressive strength
$\sigma_{\mathtt{y}}^{Lt}$	Longitudinal tensile strength
$\sigma_{\mathtt{y}}^{Tc}$	Transverse compressive strength
$\sigma_{\mathtt{y}}^{Tt}$	Transverse tensile strength
$\tau_{\mathtt{y}}^{LT}$	In-plane shear strength
$\tau_{\mathtt{y}}^{TV}$	Out-of-plane shear strength
$\varepsilon_{\mathtt{y}}^{Lc}$	Ultimate longitudinal compressive strain
$\varepsilon_{\mathtt{y}}^{Lt}$	Ultimate longitudinal tensile strain
$\varepsilon_{\mathtt{y}}^{Tc}$	Ultimate transverse compressive strain
$\varepsilon_{\mathtt{y}}^{Tt}$	Ultimate transverse tensile strain
$\varepsilon_{\mathtt{y}}^{LT}$	In-plane shear strain
$\varepsilon_{\mathtt{y}}^{TV}$	Out-of-plane shear strain
$\sigma_{\mathtt{y}}^{Lc}$	Longitudinal compressive strength

$\sigma_{\mathtt{y}}^{Lt}$	Longitudinal tensile strength
$\sigma_{\mathtt{y}}^{Tc}$	Transverse compressive strength
$\sigma_{\mathtt{y}}^{Tt}$	Transverse tensile strength
$\tau_{\mathtt{y}}^{LT}$	In-plane shear strength
$\mathcal{I}_{\mathtt{f}}$	Tsai-Wu failure index
λ	Load multiplier
λ_s	Safety factor
$\mathtt{N_d}$	Number of dimensions
$\mathcal{M}$	Number of nodes
n	Number of PGD modes
$\tilde{n}$	Arbitrary number of PGD modes
A	Arbitrary mode amplitude
`tol`	Alternated directions tolerance
β	PGD solution amplitube
$\mathtt{n_c}$	Number of compressed modes
$\mathtt{n_t}$	Number of separation terms for the elasticity tensor
$\mathtt{n_g}$	Total number of Gauss points
$\mathtt{n_r}$	Number of separation terms for the transformation matrix
$\mathcal{I}_{\mathtt{Q}}$	Quadratic term of the Tsai-Wu criterion
$\mathcal{I}_{\mathtt{L}}$	Linear term of the Tsai-Wu criterion
$\mathtt{n_Q}$	Number of modes for $\mathcal{I}_{\mathtt{Q}}$
$\mathtt{n_L}$	Number of modes for $\mathcal{I}_{\mathtt{L}}$
ξ	Greedy algorithm stopping condition
c	Cluster index or number
$\mathcal{J}$	K-means cost function value
K	Total number of clusters
ψ	Location of the centroid
D	Distance between observation and centroid
$\mathtt{N_c}$	Total number of components in data matrix
X	Arbitrary data matrix

$\tilde{\boldsymbol{\sigma}}$	Stress raw data matrix
$\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}}$	Safety factor raw data matrix
E_{max}	Maximum sum of squared error
$\mathcal{P}$	Clustering partition
$\mathtt{n_s}$	Total number of clusters in each partition $\mathcal{P}$

Vectors

$\boldsymbol{u}$	Displacement field
$\boldsymbol{\sigma}$	Stress vector
$\boldsymbol{b}$	Body forces
$\boldsymbol{u}_D$	Prescribed displacement field
$\boldsymbol{n}$	Outward normal vector
$\boldsymbol{t}_N$	Prescribed traction vector
$\boldsymbol{\varepsilon}$	Strain vector
$\boldsymbol{d}$	Nodal global displacement
$\boldsymbol{x}$	Point in 3D space
$\boldsymbol{u}_h$	Displacement trial function
$\boldsymbol{f}$	Global force vector
$\boldsymbol{\theta}$	Parameter vector

Tensors

$\boldsymbol{\mathcal{C}}$	Elasticity tensor
$\boldsymbol{\tau}$	Stress tensor
$\boldsymbol{T}$	Transformation tensor
$\boldsymbol{\mathcal{F}}$	Fourth order strength tensor
$\boldsymbol{F}$	Second order strength tensor

Chapter 1

INTRODUCTION

The present thesis applies the Proper Generalized Decomposition (PGD) framework in the parametric analysis of composite laminates. Within the computational mechanics framework, PGD provides a very efficient tool to solve the multidimensional parametric problems and obtains generalized solutions, known as *computational vademecums*. We are particularly interested in fibrous composite laminates and the goal is to ultimately optimize the orientation of the fibres to obtain enhanced mechanical properties of the structure. In the following, we present the motivation behind the current work and a brief introduction to the main ingredients of the thesis.

The motivation is presented in Section 1.1. A brief background on the main ingredients of the current thesis is in Section 1.2. We first define in a general way composite materials, their types, and list their manufacturing techniques; and then we introduce the types of structural optimization highlighting their main features. We provide a brief introduction to Model Order Reduction (MOR) techniques. We then follow by a short introduction on Data Mining (DM) and Additive Manufacturing (AM) highlighting the different technologies and their applications. In Section 1.3, we present the state of the art of the work in structural optimization of composites and the application of MOR techniques. Finally, we show the thesis scope and outline in Section 1.4. The chapter is organized as shown in Fig. 1.1.

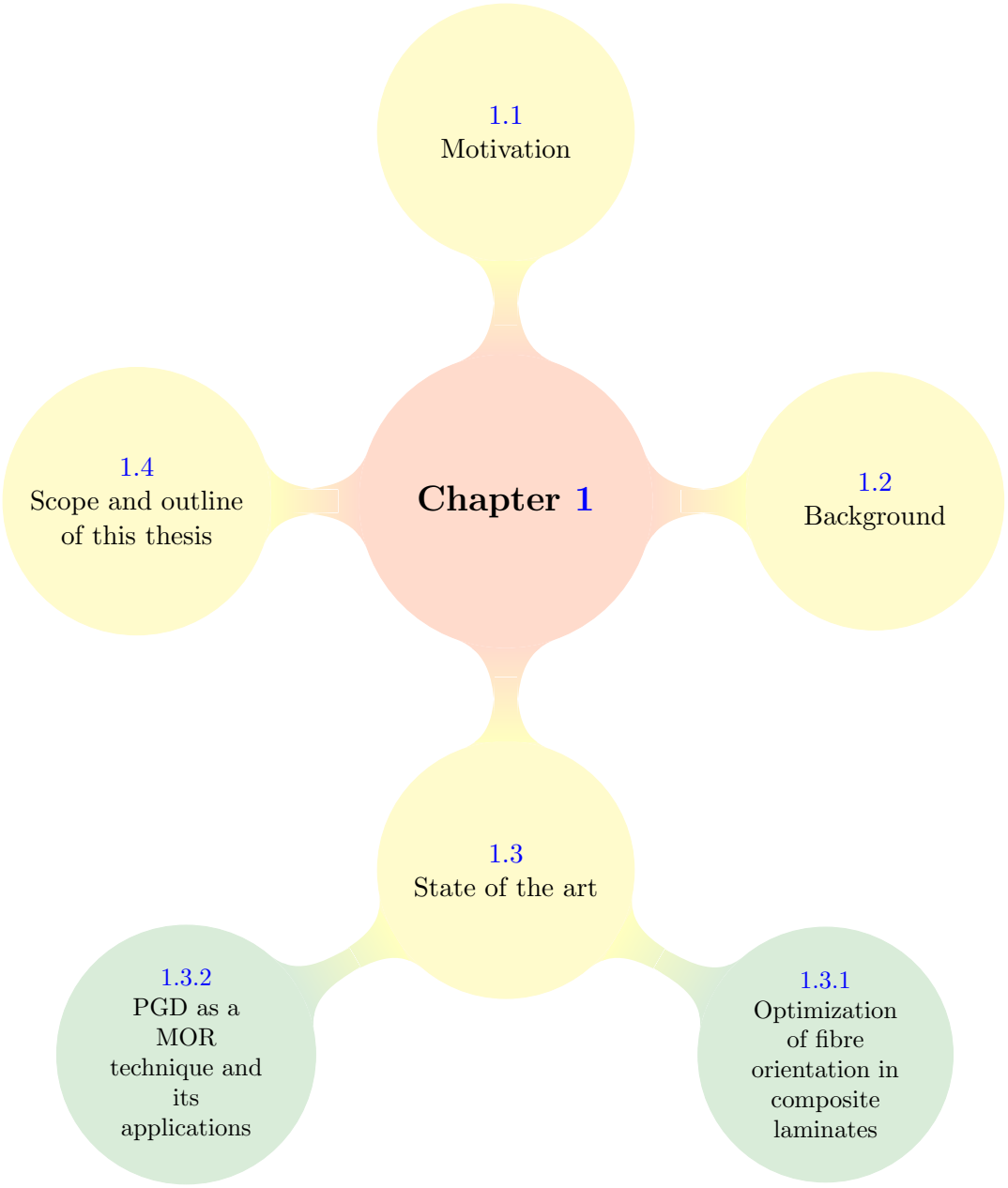


Figure 1.1: Structure of Chapter 1

1.1 Motivation

The progress of additive manufacturing allows producing tailored composite laminates, for instance with different fiber orientations in each layer or patch. Thus, the designer of the laminate has the freedom of selecting a number of parameters, e.g., angles describing the fiber orientation in each zone. In order to properly determine the optimal choice for these parameters, there is a need for modeling the mechanical behavior of a given composite laminate for any possible value of the parameters. To achieve such model, the selected parameters are considered as independent variables (extra-coordinates or extra-dimension) in the problem formulation resulting in a multidimensional problem. Generally, solving a mechanical model using mesh-based techniques in 3D is computationally expensive and at some point it could become infeasible when the problem is multidimensional. In spite of the existence of very well established theories that simplify the analysis of 3D composite laminate bodies through 2D or even into 1D structural theories, a 3D analysis is often compulsory to capture all the physics through the thickness and around the boundaries [4]. Furthermore, if the problem under consideration is an application requiring multiple queries such as optimization, inverse problems, or uncertainty quantification, the direct problem is solved numerous times increasing drastically the computational burden [5].

In the present work, we investigate the effect of the fibre orientation on the mechanical performance of fibrous composite laminates. Accordingly, an optimization technique should be applied to efficiently find the best fibre orientation in the laminate. The fibre orientation in a laminate is one of many design parameters affecting the structural performance. For example, the variation of the stacking sequence, material density or layer thickness has a direct effect on the mechanical performance of composite laminates. For this reason, it is of paramount importance to consider their optimization either individually or simultaneously [6, 7] to achieve better designs. There is a vast literature on methodologies for the optimization of the design of composite laminates; and for deeper insight the reader is referred to the review paper [8] and the references therein.

Driven by the importance of this area of research, the present thesis focuses on fibrous composite laminates and aims for quantifying the failure of the material using the Tsai-Wu failure criterion and ultimately find the best fibre orientations minimizing the failure index. These aims naturally lead us to an optimization problem that has to be solved a large number of times, corresponding to different choices of the design parameters. Thus, the computational complexity of this procedure blows up with the number of design parameters, resulting in the so-called *curse of the dimensionality* [9]. The use of a Model

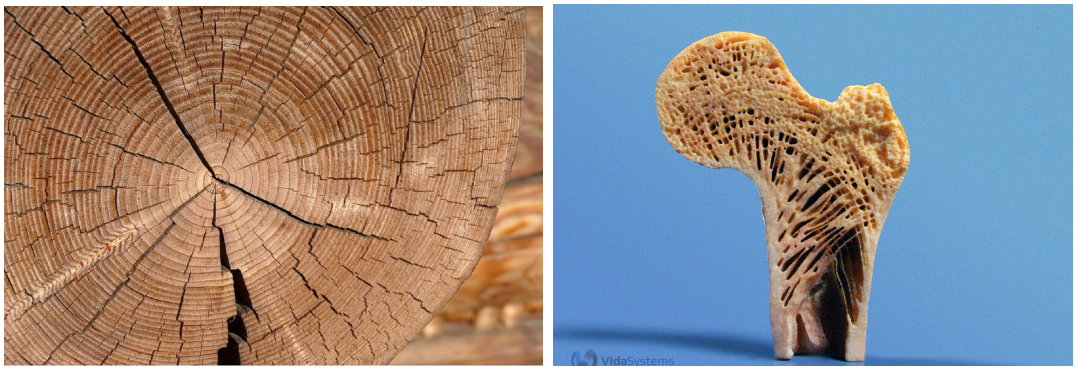
Order Reduction (MOR) technique is advocated to alleviate the mentioned computational burden. Particularly, the Proper Generalized Decomposition (PGD) method is selected as a MOR technique because it provides a solution with explicit dependence on the parameters of the problem making the optimization process straightforward. Moreover, PGD provides a generalized solution, known as a *computational vademecum*, resulting in very fast responses when browsing a particular solution. Motivated by the above, we put together different ingredients to efficiently approach the optimization of fibre orientation in composite laminates.

1.2 Background

Composite materials are widely used nowadays in numerous engineering applications such as: automotive, aerospace, biomedical, structural, to name a few. Composite materials could be defined as material possessing two or more phases bonded together [10]. The origin of composite materials is debatable between ancient Egypt and Mesopotamia, however, it is not less than 4000 years old [11]. Without a doubt, humans across different civilizations have been using composites and benefiting for example from its strength, light weight, durability, design flexibility, or even corrosion resistance. For example, it was reported in [12] that ancient Egyptians used to mix mud and straw to obtain reinforced bricks with increased strength that were used in different types of constructions.

The development of composites is progressing with the advancement of design and manufacturing technologies. Composite materials nowadays enter in most industries with different commercial applications such as machine components, thermal components; mechanical components like brakes or drive shafts; sports components, biomedical devices, to name a few. Composite materials could also be found in nature. A popular example is the trunk of a tree where the material has different orientation in the cross-section, another common example is the cross-section of the femur bone (as shown in Fig. 1.2).

A typical composite consists of reinforcement fibres and a matrix; and according to [13], composites could be classified based on the type of material used for the matrix. The main types of composites are Polymer Matrix Composites (PMC) which is the most popular type, Metal Matrix Composites (MMC), Ceramic Matrix Composites (CMC), Carbon Matrix Composites (CaMC). Reinforcement fibres could also be continuous, discontinuous, particles, braids, among numerous forms [13].



(a) Cross-section of a tree trunk (b) Cross-section of a femur bone

Figure 1.2: Different composite materials in nature

Many fibre reinforced composites are heterogeneous and anisotropic materials. Anisotropic material is a material which possesses different mechanical properties in different directions. Whereas, heterogeneous materials means that their properties vary considerably from one point to another [13]. The possibility to have different fibre orientations gives the designer the flexibility to customize the mechanical properties of the composite resulting in more durable, reliable structural components in various applications.

Designers could achieve very complex designs of mechanical components thanks to the advancement of manufacturing processes. Traditionally, composite materials are fabricated with one of the following techniques and their variations: hand lay-up, spray lay-up, Resin Transfer Molding (RTM), compression molding, injection molding, vacuum bag processing, pultrusion, and filament winding [14]. The most recent fabrication technology is the Additive Manufacturing (AM), also commercially known as 3D printing. AM is the process of building structural parts additively, by adding material in a layer-by-layer fashion. The AM technology succeeds to build rapidly very complex structural components which encourages researchers and designers to perform structural optimization of the 3D printed components and to find new materials in the pursuit of more reliable structures with a wide range of applications (example in Fig. 1.3).

Structural optimization is the study of finding the best design of a structure to fulfil a certain objective [15]. An example of the design objective to be fulfilled could be finding the minimum possible weight to strength ratio or achieving the maximum stiffness. The choice of the optimization objective is decided by the designer based on the function and conceptual design of the structure [15]. Traditionally, once the optimization objective is chosen, an iterative-intuitive trial and error procedure is followed to find the best design. Certainly, this is a very expensive and time consuming procedure resulting also in a waste of materials; and sometimes is very difficult to achieve. Luckily, with the advancement of numerical modelling and the improvement of the computational power, we are able

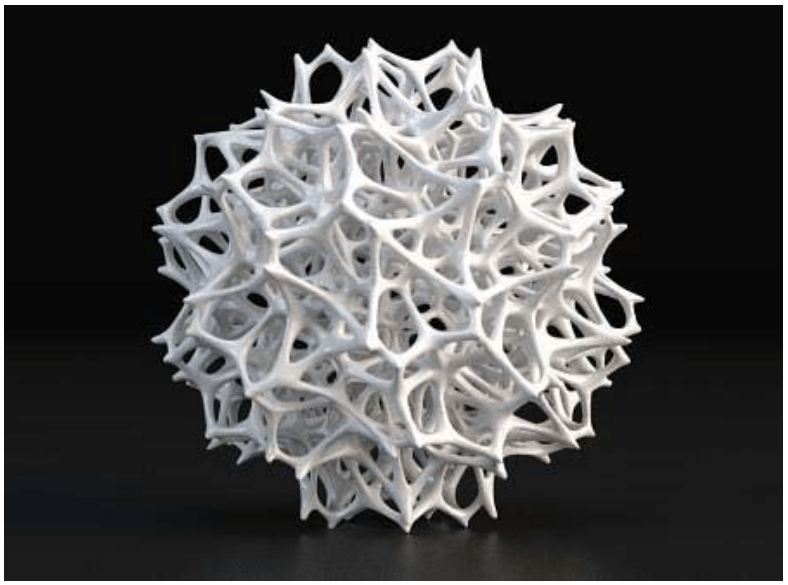


Figure 1.3: An example of a complex geometry possible with 3D printing

nowadays to undergo the optimization process virtually. In order to perform any kind of mathematical optimization, a precise mathematical problem needs to be formulated first. Given an arbitrary function f, a design variable x, and a state variable y, we show a typical optimization problem form:

$$\mathcal{SO} = \begin{cases} \text{minimize } f(x,y) \text{ w.r.t } x \text{ and } y \\ \text{subject to } \begin{cases} \text{Equilibrium constraint} \\ \text{Design constraints on } x \\ \text{Behavioral constraints on } y \end{cases} \end{cases} \tag{1.1}$$

The design variable x could represent a geometric feature of the structure, or a material parameter. A typical classification of a structural optimization problem is based on a geometric feature, and accordingly we could divide structural optimization into three classes: size, shape, and topology [15]. The optimization of materials, including fibre orientations, could be added to the just mentioned categories and therefore we could consider four layers of structural optimization [1]. For the sake of completeness, we introduce here the different types of structural optimization highlighting their main features.

Size optimization is when the design variable x in problem (1.1) is a type of structural thickness, i.e., thickness of a layer or cross section area of a membrane. While shape optimization is when the design variable x represents the shape of a part or of all of the boundary of the structural domain, i.e. shape of a hole. The topology optimization is

the most popular type of structural optimization and it consists in finding the best way to remove material while maintaining the strength of the structure as high as possible. And finally, the material optimization contribute to the changing of the orientation of the material in composites. In Fig. 1.4, we can see a schematic representation of different structural optimization layers. For further insight on structural optimization, the interested reader should refer to the work in [15, 16] and the references therein.

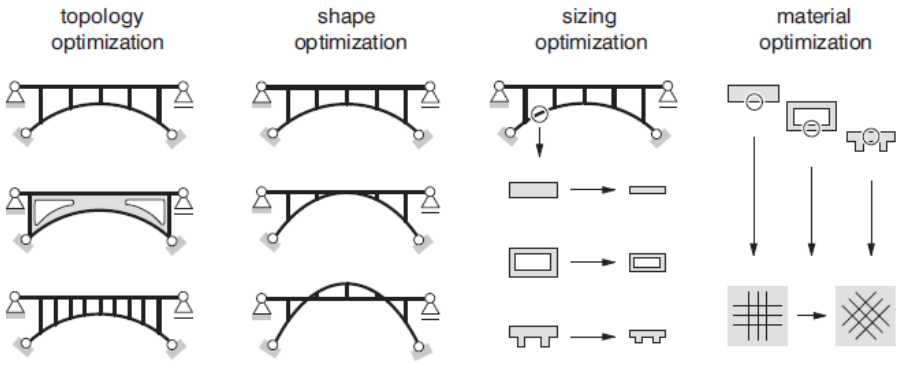


Figure 1.4: Four levels of structural optimization. Figure adapted from [1]

The change of the material orientation in composites creates new material with different mechanical properties and new characteristics. One of the characteristics of new advanced materials is that they are anisotropic. Thanks to new manufacturing technologies, such as AM, complex anisotropic materials and customized structures could be realized. However, the anisotropy of materials poses new challenges for the analysis and the optimization methods [16]. A common challenge is that the optimization of the orientation of the material is computationally very expensive. The multi-query nature of such problems consists in solving the forward problem numerous times. Moreover, the search space becomes multidimensional when we increase the number of design variables, which makes the optimization problem impractical. One way to alleviate the computational burden is to resort to Model Order Reduction (MOR) techniques.

Model Order Reduction (MOR) techniques is a family of techniques that developed enormously over the past decade. The main purpose of this family of techniques is to reduce the computational burden and the data storage needs in many fields of engineering. Traditional mesh-based numerical methods such as the Finite Element Method (FEM) are very well established for solving Partial Differential Equations (PDEs). In an optimization problem, each iteration involves a new input to the FEM model, which results in a PDE to be solved many times. The different inputs to the model, that are considered as parameters, could be geometrical, material orientation, or even boundary conditions [17]. Of course, with every new set of parameters, we end up with a full system of equations to solve and therefore resulting in a huge increase in the computational cost. For the just mentioned challenges, one would resort to MOR techniques to reduce the computational

cost.

Another ingredient in the current thesis is Data Mining (DM). The coupling between Computational Mechanics (CM) models and DM techniques has recently gained a lot of popularity [18, 19]. DM is the core of discovering knowledge in large complex databases. The knowledge discovery process is achieved by employing algorithms to explore the data and develop models to extract unknown patterns. The term Data Mining has been coined to explain the process of going through big databases looking for informative patterns and connections between data. With the exponential increase of data availability, DM field is of paramount importance to keep up the processing level of this data [20].

Predictive models are one of the most popular tasks of DM. It involves processing history data, identifying patterns in the data, and then use an already built model to predict the future. An example of a predictive model is the weather forecast. Another important task of DM is the segmentation of data. It consists in collecting items having similar features into mutually exclusive groups [21]. In the current work we are interested in segmentation of data through the so-called unsupervised clustering algorithms. According to [21], clustering methods could be divided into three main groups, unsupervised, semi-supervised, and supervised:

(1) **Unsupervised:** the aim of unsupervised clustering is to maximize the within cluster similarity and minimize the intercluster similarity using a metric measure without any information about the output. The most popular algorithms used in segmentation are the K-means and the hierarchical clustering.
(2) **Semi-supervised:** this type of clustering uses additional information to guide the clustering algorithm and improve the results.
(3) **Supervised:** in the presence of class labels and with a priori knowledge of the types of output we want, clustering could be considered supervised.

Inspired by the recent work of Alaimo *et al.* in [22], we would like to apply such techniques to be able to improve the methodology proposed in the present thesis.

The final ingredient which motivates the current work is the Additive Manufacturing (AM) technology. AM refers to a series of technologies that aims at building products by adding material rather than the traditional subtractive manufacturing processes (e.g. drilling, milling, etc). The AM technology has been around for more than two decades now and it is growing rapidly due to its great potential in the industry. AM technologies, also referred to commercially as 3D printing, contribute to many fields of science and engineering such as: automotive, medical and healthcare, aerospace and defense,

consumer products, to name a few.

AM technologies possess many different advantages against traditional manufacturing techniques that makes it very attractive to designers. First of all and most importantly, it could achieve very complex geometries with very high accuracy giving flexibility to designers which is not always possible with traditional techniques. The ability to produce complex designs opens the door for new customized materials to be investigated by researchers. Second, it reduces the waste of material since the process is additive, unlike the subtractive manufacturing techniques where there is a huge loss of material. It is a user-friendly technology that could be in everyone's home; with minimum computer skills, an untrained person could use 3D printing. Finally, it is fast to produce a part, and depending on the AM technology used, it requires minor post-processing enabling on-demand manufacturing.

1.3 State of the art

1.3.1 Optimization of fibre orientation in composite laminates

The interest in obtaining new material is increasing everyday. Coupled with the advancement of AM technologies, the realization of composite laminates with complex material orientation is possible [23]. It was shown in many studies that the optimization of the failure of the material [24], stiffness [25], or strain energy [26] affects the overall material properties. Many recent works have been dedicated to investigate the optimization of different aspects of composite laminates. The outstanding review paper by Nikbakt *et al.* [8] summarizes to a good extent the different kinds of composite structures and the methods employed for their optimization. We could classify the process of optimizing the material orientation in composite laminates based on the objective function used or the optimization algorithm used. Early work by Pedersen has been dedicated to solve the optimization problem of the orientation of orthotropic material analytically using a strain based objective function [27]. Pedersen continued his work and devised a FEM and optimization procedure to solve the aforementioned problem and also to solve the thickness-orientation optimization problem [26, 28]. The early work inspired Thomsen to add topology optimization for composite discs using Pedersen's method [29]. Another approach was proposed by Luo *et al.* that is based on energy to determine analytically the optimal orientation of orthotropic material [25]. The early work paved the way for more researchers to investigate the optimization of fibre orientations using different

approaches. More recent work by Huang *et al.* involving the optimization of fibre orientation using a load bearing approach. Particularly, they used the famous Tsai-Wu criterion as the objective function [24]. It was followed by the work of Groenwold and Haftka on the optimization of different failure criteria, which is considered as a main reference for the current thesis [30]. The work by Bruynmeel consisted in optimizing the fibre orientation using strength based criteria, such as Tsai-Hill and Tsai-Wu, showing that a direct parameterization of the problem in the design variables is favorable over the parameterization using lamination parameters where the space is not completely known [31]. Bruynmeel extended his work by providing a more complete picture of structural optimization. He argued that the optimization of the fibres in a non-homogeneous domain is very sensitive to the initial guess in gradient based methods and that one could end up with a local solution. Bruynmeel also reported that the more the number of design variables increases the more the optimization problem becomes expensive, especially if the optimization method employed is a non-deterministic method to obtain global solutions [32].

More recent work done by Hwang *et al.* addressing the optimization of fibre orientation in each layer using the Genetic Algorithm (GA) and it was reported that the optimization algorithm behaves well in such problems [33]. It was shown by Li *et al.* in a recent research that a hybrid optimization method consisting in the genetic algorithm and the particle swarm optimization efficiently finds the optimal solution [34]. Another recent work combining topology optimization with fibre orientation optimization by minimizing the compliance. They use gradient-based method which leads to local solutions. Moreover, solving the system and the computation of the sensitivities every iteration could lead to a computationally expensive problem depending on the size of the system and the number of design variables [23]. A recent study by Diniz *et al.* focuses on the design optimization using the Tsai-Wu failure criterion as the objective function coupled with Artificial Neural Networks (ANN) [35]. Another work considering fibre orientation optimization by minimizing the compliance in hyperelastic material is presented in [36]. It was shown that it is a good practice to employ a filter to obtain a good continuity of the optimal fibres. The work done by Shen *et al.* is very competitive and it shows the optimization of fibre orientation using the compliance as an objective function. However, it is reported that the solution is easily affected by the initial guess since the work employs a gradient-based method to obtain the optimal fibre orientation. They overcome this issue by providing the gradient-based method with the principle directions as initial guess. Moreover, in each iteration they solve the complete system, which could become computationally challenging if the system is large [37].

1.3.2 PGD as a MOR technique and its applications

The use of standard techniques to solve an optimization problem is computationally expensive as it involves solving the full model a large number of times for different values of the design variables (i.e. model parameters). Thus, one would consider the application of MOR techniques to reduce the computational burden of such problems. A very well established MOR class is the Reduced Basis (RB) method. RB is based on taking snapshots of the system, which are a collection of full solutions of the system at different values of a parametric set. The snapshots are then combined together to form a basis for the problem [38–41]. The building of the basis is a straightforward procedure, whereas the main challenge remains in the choice of the values of the parametric set where to take the snapshots and also in choosing the number of snapshots used to build the basis. The most common strategy to build the reduced basis space is the Proper Orthogonal Decomposition (POD). Depending on the field, POD is also known as Principal Component Analysis (PCA), Singular Value Decomposition (SVD), or Karhunen-Loève transform [42, 43]. POD aims to generate orthonormal basis that are optimal based on a Galerkin (or least-square) projection resulting in the elimination of any redundancies in the formed basis functions representing the system [41]. An error estimator is usually employed to assess the quality of the basis obtained, the interested reader could refer to [38, 44, 45] for a deeper insight on the topic.

Another MOR technique, which developed enormously over the last decade, is the Proper Generalized Decomposition (PGD) [5, 9, 46–51]. Unlike RB and POD, the PGD is a *priori* MOR technique, i.e., it does not rely on the approximate solution of the full problem. PGD is most useful in multi-variable parametric problems such as optimization problems, inverse identification, and uncertainty quantification, where the forward problem is solved a large number of times [47]. In a nutshell, the PGD aims at two things in a multi-variable problem. First, it aims at considering all the design variables in the problem as extra-coordinate, making the model explicit in the parameters and thus resulting in an explosion in the computational cost known as *curse of dimensionality*. Second, it aims at representing the solution of the parametric PDE in a separated fashion to alleviate the computational burden. There are two distinct phases when applying PGD, the first is the *offline* phase and the second is the *online* phase. The offline phase takes important computational resources and results in a generalized solution known as *computational vademecum*. The computational vademecum acts like a modern virtual chart having all the possible solutions of the problem for any value of the set of parameters. This is particularly useful for multi-query problems, as the browsing of the solution for a value of the parametric set happens in seconds, which occurs in the online phase, opening the door for real-time simulations. Thus, an optimization problem becomes a simple post-process especially that the sensitivities are explicit in the design parameters.

PGD is applied in many fields. The work by Bognet *et al.* addresses plates and shells geometries providing 3D solutions with 2D computational complexity [4]. The same problem was addressed by Vidal *et al.* choosing a piecewise fourth-order Lagrange polynomial instead of a linear piecewise polynomial [52, 53]. The work by Leygue and Verron took the first steps towards structural optimization [54]. It was followed a few years later by the work of Ammar *et al.* applying PGD to shape optimization problems [55]. A very recent work by Sibileau *et al.* provides parametric solutions for lattice structures [56]. PGD is also applied in computational rheology [57], in solving Navier-Stokes equation for water agitation in harbours [58], in solving the Stokes problem [59], power supply systems [60], real-time monitoring of thermal processes [61], and damage models [62]. Many works were dedicated to solving inverse problems and uncertainty quantification in different fields such as: parameter identification in geophysics problems [63] and in heat transfer problems [64], and uncertainty quantification in physics [65]. Finally, the work by Courard *et al.* took the first steps in incorporating a PGD vademecum in a non-intrusive way as a part of an engineering process (e.g. optimization process) [66]. A very recent work by Díez *et al.* presented an algorithmic approach to high-dimensional tensor separation and they extended the PGD approach to obtain a set of tools to operate with multidimensional data which is applied in the current thesis [50, 51].

1.4 Scope and outline of this thesis

The objective of the current thesis is to present a new computational tool for the numerical analysis of 3D fibrous composite laminates problems with the goal of optimizing the fibres orientations. This is particularly useful for additive manufacturing applications. The used in-house algorithm is based on solving a set of algebraic equations where the unknown is a generalized vector of deformations of the composite laminate. The generalized solution obtained from the solver, stored in a computational vademecum, is explicit in the design variables which is very useful for the optimization problem. The code is then coupled with a post-process algorithm that evaluates the stresses and the failure index of the structure obtaining a failure vademecum. The failure vademecum is then inserted in an optimization algorithm to find the best fibre orientation in the structure. The non-convex nature of the optimization objective motivates the use of the evolutionary optimization algorithm, the Genetic Algorithm (GA). The present work focuses only on linear elastic constitutive model as a preliminary analysis step. Moreover, the system is spatially discretized using the Finite Element Method (FEM). Finally, the model is also supplemented with a data analysis algorithm that finds the best domain decomposition strategy for the PGD problem and hence enhancing the methodology.

For clarity we outline the present thesis below

- **Chapter 2**: *Problem statement.*
  This chapter introduces the problem statement including the governing equations for linear elasticity in 3D, a brief introduction for the different failure criteria, and the optimization problem to be solved to find the best fibre orientation in the composite laminate.

- **Chapter 3**: *Proper Generalized Decomposition (PGD) framework.*
  This chapter presents the PGD framework with a brief introduction, then introduces the *encapsulated* PGD approach [51], and shows how the PGD tools are adapted to the problem in hand. It also presents the separation process of the PGD input and the sensitivities computation.

- **Chapter 4**: *Numerical examples.*
  In this chapter, a series of numerical examples are simulated to demonstrate the capabilities of the methodology. Results are first compared with FEM and then a more challenging example is simulated with a higher number of parameters.

- **Chapter 5**: *Domain decomposition using data analysis techniques for efficient PGD parameterization.*
  In this chapter, we introduce data analysis techniques with the aim to enhance the proposed methodology. The idea in this chapter is to obtain a clustering optimization algorithm that automatically finds the best domain partitioning (i.e. parameterization) strategy for the PGD solver.

- **Chapter 6**: *Concluding remarks.*
  Finally, we wrap up with some concluding remarks on the work done in the presented thesis. Future potential directions of research are also introduced.

Chapter 2

PROBLEM STATEMENT

2.1 Preliminaries

The current chapter presents the problem statement and the methodology applied in this thesis. Firstly, the governing equations used for the numerical analysis of the problems along with the constitutive law are shown in Section 2.2. Then the parametrization of the problem is explained in Section 2.3. The description of various popular failure criteria is briefly introduced along with a deeper explanation of the Tsai-Wu criterion in Section 2.4. Finally, Section 2.5 provides a short description of the optimization problem in hand and the techniques that will be used are briefly explained. The layout of the chapter is shown in Fig. 2.1.

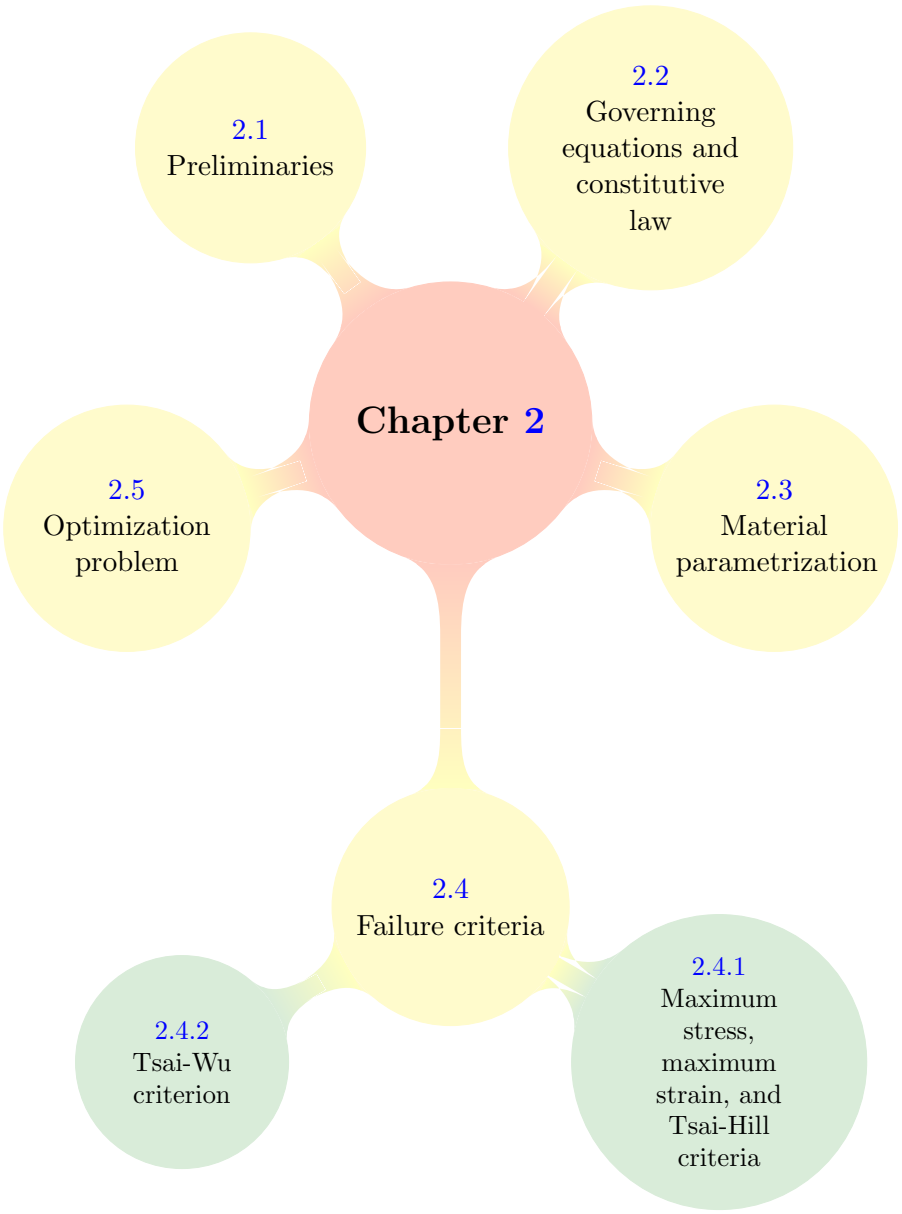


Figure 2.1: Structure of Chapter 2

2.2 Governing equations and constitutive law

Given a 3D domain $\Omega \subset \mathbb{R}^3$, the linear elasticity problem consists in solving the following Boundary Value Problem (BVP), here presented in its strong form:

find the displacement $\boldsymbol{u}$ satisfying the equilibrium equation and boundary conditions,

$$\begin{cases} \boldsymbol{\nabla}_S^{\mathsf{T}}\boldsymbol{\sigma} + \boldsymbol{b} = \boldsymbol{0} & \text{in } \Omega \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \Gamma_D \\ \boldsymbol{n}^{\mathsf{T}}\boldsymbol{\sigma} = \boldsymbol{t}_N & \text{on } \Gamma_N \end{cases} \tag{2.1}$$

with,

$$\begin{aligned} \boldsymbol{\sigma} &= \mathcal{C}\boldsymbol{\varepsilon} \\ \boldsymbol{\varepsilon} &= \boldsymbol{\nabla}_S \boldsymbol{u}, \end{aligned} \tag{2.2}$$

where $\boldsymbol{\nabla}_S$ is a 6×3 symmetric gradient matrix operator (described in [67]), $\boldsymbol{\sigma}$ is the stress field, $\boldsymbol{b}$ is the body forces vector; $\boldsymbol{u}_D$ is the displacement field prescribed on the Dirichlet boundary, $\boldsymbol{t}_N$ is the prescribed the traction applied on the Neumann boundary, $\boldsymbol{n}$ is the 6×3 matrix representation of the normal (analogous to $\boldsymbol{\nabla}_S$ operator); $\mathcal{C}$ is the elasticity tensor, and $\boldsymbol{\varepsilon}$ is the strain field. The stresses and strains are expressed in the engineering Voigt's notation (vectors of six components); and using the same notation, the elasticity tensor is expressed as a 6×6 matrix. Thus, for orthotropic materials, the constitutive relation in (2.2) reads,

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ & C_{22} & C_{23} & 0 & 0 & 0 \\ & & C_{33} & 0 & 0 & 0 \\ & \text{Sym.} & & C_{44} & 0 & 0 \\ & & & & C_{55} & 0 \\ & & & & & C_{66} \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{bmatrix}. \tag{2.3}$$

Following the derivations in [67] to obtain the weak form, we first define the admissible weight function and trial solution. Then we premultiply the equilibrium equation and the boundary conditions by the weight function and integrate over the domain Ω. This results in the following weak form after applying the divergence theorem and integrating by parts,

$$\int_\Omega (\boldsymbol{\nabla}_{\mathsf{S}}\boldsymbol{w})^{\mathsf{T}}\boldsymbol{\sigma}\, d\Omega = \int_{\Gamma_N} \boldsymbol{w}^{\mathsf{T}}\boldsymbol{t} d\Gamma + \int_\Omega \boldsymbol{w}^{\mathsf{T}}\boldsymbol{b} d\Omega, \tag{2.4}$$

where $\boldsymbol{w}$ is the weight function and it is chosen in a way such that it vanishes on the Dirichlet boundary. We then substitute the constitutive law in Eq. (2.2) in the weak form yielding the following,

solve for $\boldsymbol{u} \in U$ such as,

$$\int_\Omega (\boldsymbol{\nabla}_{\mathsf{S}}\boldsymbol{w})^{\mathsf{T}}\boldsymbol{\mathcal{C}}\boldsymbol{\nabla}_{\mathsf{S}}\boldsymbol{u}\, d\Omega = \int_{\Gamma_N} \boldsymbol{w}^{\mathsf{T}}\boldsymbol{t} d\Gamma + \int_\Omega \boldsymbol{w}^{\mathsf{T}}\boldsymbol{b} d\Omega \qquad \forall \boldsymbol{w} \in U_o$$

$$\text{where,} \quad U = \{\boldsymbol{u} | \boldsymbol{u} \in \left(H_1(\Omega)\right)^3,\ \boldsymbol{u} = \boldsymbol{u}_D \quad \text{on } \Gamma_D\},\ U_0 = \{\boldsymbol{w} | \boldsymbol{w} \in \left(H_1(\Omega)\right)^3,\ \boldsymbol{w} = 0 \quad \text{on } \Gamma_D\}. \tag{2.5}$$

Remark 1: We define the trial solution space as $U = \{\boldsymbol{u} | \boldsymbol{u} \in \left(H_1(\Omega)\right)^3,\ \boldsymbol{u} = \boldsymbol{u}_D \quad \text{on} \quad \Gamma_D\}$ and the test function space as $U_0 = \{\boldsymbol{w} | \boldsymbol{w} \in \left(H_1(\Omega)\right)^3,\ \boldsymbol{w} = 0 \quad \text{on} \quad \Gamma_D\}$, where $H_1(\Omega)$ is a Hilbert space containing square-integrable functions with derivatives that are also square-integrable.

We denote the total number of nodes per element as $\mathtt{n_{ne}}$ and the total number of nodes in the domain as $\mathtt{n_{nt}}$ whereas the whole domain has a total number of elements $\mathtt{n_{el}}$. For a 3D domain, there are three degrees of freedom per node and the global nodal displacement vector reads,

$$\boldsymbol{d} = [u_{x1} \ \ u_{y1} \ \ u_{z1} \ \ u_{x2} \ \ u_{y2} \ \ u_{z2} \ \ \cdots \ \ u_{x\mathtt{n_{nt}}} \ \ u_{y\mathtt{n_{nt}}} \ \ u_{z\mathtt{n_{nt}}}]^{\mathsf{T}}.$$

In the finite dimensional context, the continuous displacement field $\boldsymbol{u}$ at a given point $\boldsymbol{x} = [x \ y \ z]^{\mathsf{T}}$ is approximated by the trial function $\boldsymbol{u}_h$ and is expressed as follows within an element domain Ω^e with $\mathtt{n_{edof}}$ element degrees of freedom,

$$\begin{aligned} \boldsymbol{u}_{\boldsymbol{h}}^{\boldsymbol{e}}(\boldsymbol{x}) &= \boldsymbol{N}^{\boldsymbol{e}}(\boldsymbol{x})\boldsymbol{d}^{\boldsymbol{e}} \qquad \boldsymbol{x} \in \Omega^e \\ \boldsymbol{w}_{\boldsymbol{h}}^{\boldsymbol{e}}(\boldsymbol{x})^{\mathsf{T}} &= \mathbf{w}^{e\mathsf{T}}\boldsymbol{N}^{\boldsymbol{e}}(\boldsymbol{x})^{\mathsf{T}} \qquad \boldsymbol{x} \in \Omega^e, \end{aligned} \tag{2.6}$$

where $\boldsymbol{N}^e$ is a $3 \times \mathtt{n_{edof}}$ elemental shape function matrix having the following form,

$$\boldsymbol{N}^{\boldsymbol{e}} = \begin{bmatrix} N_1^e & 0 & 0 & N_2^e & 0 & 0 & \cdots & N_{\mathtt{n_{ne}}}^e & 0 & 0 \\ 0 & N_1^e & 0 & 0 & N_2^e & 0 & \cdots & 0 & N_{\mathtt{n_{ne}}}^e & 0 \\ 0 & 0 & N_1^e & 0 & 0 & N_2^e & \cdots & 0 & 0 & N_{\mathtt{n_{ne}}}^e \end{bmatrix},$$

where $\boldsymbol{u}_{\boldsymbol{h}}^{\boldsymbol{e}}$ is the 3×1 element trial solution and $\boldsymbol{d}^e$ is a $\mathtt{n_{edof}} \times 1$ element nodal displacements vector. The integral over the whole domain in (2.5) could be computed as a sum of integrals over element domains Ω^e as follows:

$$\sum_{e=1}^{\mathtt{n_{el}}} \left\{ \int_{\Omega^e} (\boldsymbol{\nabla}_{\mathrm{S}} \boldsymbol{w}_{\boldsymbol{h}}^{\boldsymbol{e}})^{\mathsf{T}} \boldsymbol{\mathcal{C}} \boldsymbol{\nabla}_{\mathrm{S}} \boldsymbol{u}_{\boldsymbol{h}}^{\boldsymbol{e}} \, d\Omega - \int_{\Gamma_N^e} \boldsymbol{w}_h^{e\mathsf{T}} \boldsymbol{t} \, d\Gamma - \int_{\Omega^e} \boldsymbol{w}_{\boldsymbol{h}}^{e\mathsf{T}} \boldsymbol{b} \, d\Omega \right\} = 0. \tag{2.7}$$

It is convenient now to follow the same procedure and express the strain field in terms of the shape functions,

$$\boldsymbol{\varepsilon}_{\boldsymbol{h}}^{\boldsymbol{e}}(\boldsymbol{x}) = \boldsymbol{\nabla}_{\mathrm{S}} \boldsymbol{u}_{\boldsymbol{h}}^{\boldsymbol{e}}(\boldsymbol{x}) = \boldsymbol{\nabla}_{\mathrm{S}} \boldsymbol{N}^{\boldsymbol{e}}(\boldsymbol{x}) \boldsymbol{d}^{\boldsymbol{e}} = \boldsymbol{B}^{\boldsymbol{e}}(\boldsymbol{x}) \boldsymbol{d}^{\boldsymbol{e}} \qquad \boldsymbol{x} \in \Omega^e, \tag{2.8}$$

where $\boldsymbol{B}^e$ is the $6 \times \mathtt{n_{edof}}$ strain-displacement matrix containing the symmetric gradient of the shape functions and has the following structure

$$\boldsymbol{B^e} \equiv \boldsymbol{\nabla}_{\mathrm{S}} \boldsymbol{N^e} = \begin{bmatrix} \frac{\partial N_1^e}{\partial x} & 0 & 0 & \frac{\partial N_2^e}{\partial x} & 0 & 0 & \cdots & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1^e}{\partial y} & 0 & 0 & \frac{\partial N_2^e}{\partial y} & 0 & \cdots & 0 & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1^e}{\partial z} & 0 & 0 & \frac{\partial N_2^e}{\partial z} & \cdots & 0 & 0 & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial z} \\ 0 & \frac{\partial N_1^e}{\partial z} & \frac{\partial N_1^e}{\partial y} & 0 & \frac{\partial N_2^e}{\partial z} & \frac{\partial N_2^e}{\partial y} & \cdots & 0 & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial z} & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial y} \\ \frac{\partial N_1^e}{\partial z} & 0 & \frac{\partial N_1^e}{\partial x} & \frac{\partial N_2^e}{\partial z} & 0 & \frac{\partial N_2^e}{\partial x} & \cdots & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial z} & 0 & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial x} \\ \frac{\partial N_1^e}{\partial y} & \frac{\partial N_1^e}{\partial x} & 0 & \frac{\partial N_2^e}{\partial y} & \frac{\partial N_2^e}{\partial x} & 0 & \cdots & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial y} & \frac{\partial N_{\mathtt{n_{ne}}}^e}{\partial x} & 0 \end{bmatrix},$$

and $\boldsymbol{\varepsilon_h^e}$ is the 6×1 elemental strain field at point $\boldsymbol{x}$. Similarly, the derivatives of the test function in Eq. (2.7) could be written in terms of the strain-displacement matrix as follows,

$$(\boldsymbol{\nabla}_{\mathrm{S}} \boldsymbol{w_h^e})^{\mathsf{T}} = (\boldsymbol{\nabla}_{\mathrm{S}} \boldsymbol{N^e} \mathbf{w^e})^{\mathsf{T}} = (\boldsymbol{B^e} \mathbf{w^e})^{\mathsf{T}} = \mathbf{w}^{e\mathsf{T}} \boldsymbol{B}^{e\mathsf{T}}. \tag{2.9}$$

Substituting Eq. (2.6), Eq. (2.8) and Eq. (2.9) in Eq. (2.7) yields,

$$\sum_{e=1}^{\mathtt{n_{el}}} \left\{ \int_{\Omega^e} (\mathbf{w}^{e\mathsf{T}} \boldsymbol{B}^{e\mathsf{T}} \boldsymbol{\mathcal{C}} \boldsymbol{B^e} \boldsymbol{d^e}\, d\Omega - \int_{\Gamma_N^e} \mathbf{w}^{e\mathsf{T}} \boldsymbol{N}^{e\mathsf{T}} \boldsymbol{t}\, d\Gamma - \int_{\Omega^e} \mathbf{w}^{e\mathsf{T}} \boldsymbol{N}^{e\mathsf{T}} \boldsymbol{b}\, d\Omega \right\} = 0. \tag{2.10}$$

Recalling that $\boldsymbol{d^e} = \boldsymbol{L^e d}$ and $\mathbf{w}^{e\mathsf{T}} = \mathbf{w}^{\mathsf{T}} \boldsymbol{L}^{e\mathsf{T}}$ where $\boldsymbol{d}$ is the global displacements vector, $\mathbf{w}$ is the global nodal vector of weight functions and $\boldsymbol{L^e}$ is a $\mathtt{n_{edof}} \times \mathtt{n_d}$ gather matrix that extracts the element displacements vector from $\boldsymbol{d}$. Note that the matrix $\boldsymbol{L^e}$ is a matrix that consists of only ones and zeros. Accordingly the weak form is then rearranged and results in the following,

$$\mathbf{w}^{\mathsf{T}} \left\{ \sum_{e=1}^{\mathtt{n_{el}}} \left[\int_{\Omega^e} \boldsymbol{L}^{e\mathsf{T}} \boldsymbol{B}^{e\mathsf{T}} \boldsymbol{\mathcal{C}} \boldsymbol{B^e} \boldsymbol{L^e d}\, d\Omega - \int_{\Gamma_N^e} \boldsymbol{L}^{e\mathsf{T}} \boldsymbol{N}^{e\mathsf{T}} \boldsymbol{t}\, d\Gamma - \int_{\Omega^e} \boldsymbol{L}^{e\mathsf{T}} \boldsymbol{N}^{e\mathsf{T}} \boldsymbol{b}\, d\Omega \right] \right\} = 0, \tag{2.11}$$

where,

$$\boldsymbol{K^e} = \int_{\Omega^e} \boldsymbol{B}^{e\mathsf{T}} \boldsymbol{\mathcal{C}} \boldsymbol{B^e}\, d\Omega \tag{2.12}$$

$$\boldsymbol{f^e} = \int_{\Omega^e} \boldsymbol{N}^{e\mathsf{T}} \boldsymbol{b}\, d\Omega + \int_{\Gamma_N \bigcap \bar{\Omega}^e} \boldsymbol{N}^{e\mathsf{T}} \boldsymbol{t}\, d\Gamma. \tag{2.13}$$

Substituting Eq. (2.12) and Eq. (2.13) in Eq. (2.11), the final weak form has the following form,

$$\mathbf{w}^{\mathsf{T}} \underbrace{\left\{ \sum_{e=1}^{\mathtt{n_{el}}} \left[\boldsymbol{L}^{e\mathsf{T}} \boldsymbol{K}^{e} \boldsymbol{L}^{e} \boldsymbol{d} - \boldsymbol{L}^{e\mathsf{T}} \boldsymbol{f}^{e} \right] \right\}}_{\text{residual vector}} = 0. \tag{2.14}$$

Since $\mathbf{w}$ has to be arbitrary, then the residual vector has to be equal to zero to satisfy Eq. (2.14). Therefore, the global system of equations reads

$$\boldsymbol{K}\boldsymbol{d} = \boldsymbol{f}, \tag{2.15}$$

where,

$$\boldsymbol{K} := \sum_{e=1}^{\mathtt{n_{el}}} \boldsymbol{L}^{e\mathsf{T}} \boldsymbol{K}^{e} \boldsymbol{L}^{e} \text{ and } \boldsymbol{f} := \sum_{e=1}^{\mathtt{n_{el}}} \boldsymbol{L}^{e\mathsf{T}} \boldsymbol{f}^{e}. \tag{2.16}$$

2.3 Material parametrization

Material types differ according to the symmetries they possess. The material in its most generic form is called anisotropic. Anisotropic material possesses different physical properties in all directions, as opposed to the isotropic material that possesses the same properties in all directions. For a linear elastic anisotropic material, the stress tensor and the strain tensor are linearly related through the elasticity tensor. The elasticity tensor $\mathbb{C}$ is a fourth order tensor such that in the most generic form has 81 independent components. Given the symmetry of the stress and strain tensors, that is called the *minor* symmetry, a certain symmetry also holds for the elasticity tensor and, hence, its number of independent components reduces from 81 to 36. There exists also a symmetry of the elasticity tensor called the *major* symmetry which reduces the number of independent components from 36 to 21. Certainly the fourth order tensor could be represented by a 6×6 matrix $\boldsymbol{\mathcal{C}}$, while the stress and strain tensors could be represented as 6×1 vectors using the engineering Voigt's notation [68]. Other types of material symmetries reduce further the number of independent components in the elasticity tensor. A material having independent properties in three mutually orthogonal directions is called an orthotropic material. It is a subset of anisotropic material and its symmetry results in the reduction of the number of independent components in the elasticity tensor from 21 to 9 components. Fig. 2.2 shows a piece of wood that is a great example of orthotropic material in the nature, where the three mutually orthogonal directions at a point are the longitudinal direction along the fibres, the radial direction perpendicular to the ring-like structure in wood, and finally the tangential direction tangent to the ring-like structure [2].

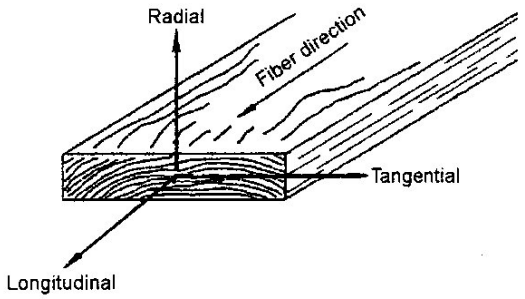


Figure 2.2: Independent directions in orthotropic material. Figure from [2] with permission.

Another material symmetry is when the material has its physical properties symmetric about one direction that is perpendicular to a plane of isotropy as shown in Fig. 2.3. This material is called transversely isotropic material and the number of independent components in its elasticity tensor is five. Finally the isotropic material, such as glass, has the same properties in all directions and the elasticity tensor has two independent components.

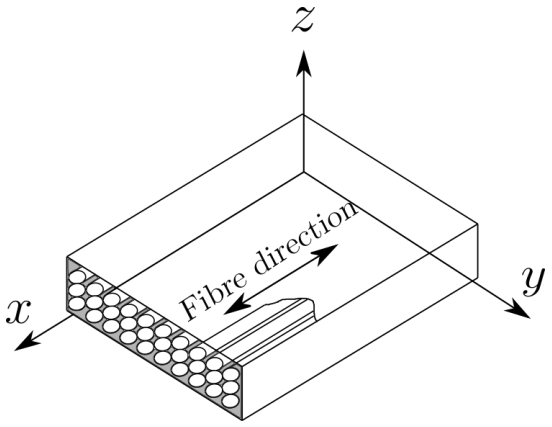


Figure 2.3: Transversely isotropic material with yz-plane as the plane of isotropy

It is assumed in the following that fibres always lie in planes parallel to the $1-2$ plane in a $\{O, 1, 2, 3\}$ material coordinate system (as shown in Fig. 2.4), where direction 1 is always the fibres' longitudinal direction. The existence of the fibres is modelled using a transversely isotropic material assumption. The reference coordinate system (or global axes) is denoted by $\{O, x, y, z\}$ and axis z coincides with axis 3, consequently, plane $\{O, x, y\}$ coincides with plane $\{O, 1, 2\}$, as shown in Fig. 2.4. The angle θ is the orientation of the family of fibres belonging to the material coordinate system with respect to the global coordinate system. Note that, the domain could be divided into many subdomains (layers or patches), hence, θ could take a specific value for each specific subdomain with respect to the global coordinate system.

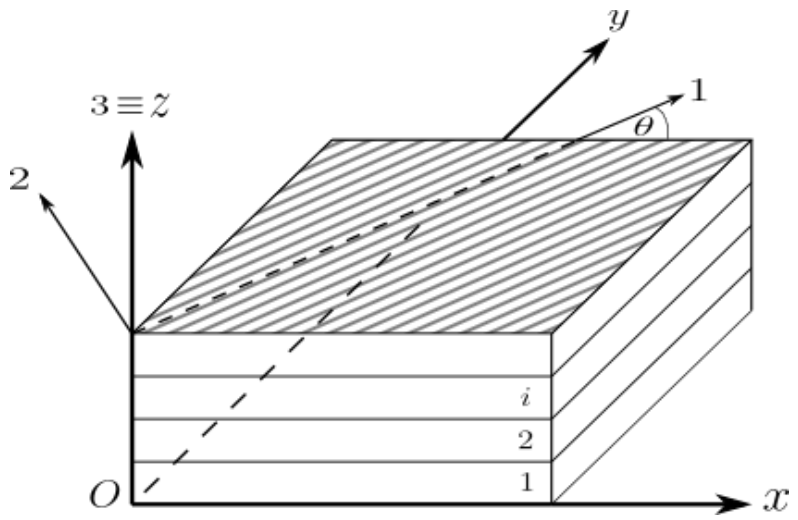


Figure 2.4: Global coordinate system {O,x,y,z} and material coordinate system {O,1,2,3}

The elasticity tensor $\boldsymbol{\mathcal{C}}$ in (2.2) when expressed in the material axes $\{O, 1, 2, 3\}$ in terms of the Young's moduli, shear moduli, and Poisson's ratios, is denoted as $\boldsymbol{\mathcal{C}}_0$. Since our problem is defined with respect to the global axes $\{O, x, y, z\}$ and we aim to have different fiber orientations in different parts of the domain, $\boldsymbol{\mathcal{C}}_0$ has to be also expressed in the global reference. Thus, the question is how to represent $\boldsymbol{\mathcal{C}}(\theta)$ as a function of $\boldsymbol{\mathcal{C}}_0$ and θ.

The transformation of $\boldsymbol{\mathcal{C}}_0$ to $\boldsymbol{\mathcal{C}}(\theta)$ is a result of the transformation of stresses and strains from material to global axes. These transformations are well established in the literature [69, 70] and make use of a transformation matrix $\boldsymbol{T}(\theta)$. The matrix $\boldsymbol{T}$ is applied to stresses and strains as follows

$$\begin{aligned} \boldsymbol{\sigma}_{123} &= \boldsymbol{T}(\theta)\boldsymbol{\sigma}_{xyz} \\ \boldsymbol{\varepsilon}_{123} &= \boldsymbol{T}^{-\mathsf{T}}(\theta)\boldsymbol{\varepsilon}_{xyz}, \end{aligned} \tag{2.17}$$

where the subscript "123" refers to the material axes and "xyz" refers to the global axes. We could then define the matrix $\boldsymbol{T}$ as

$$\boldsymbol{T}(\theta) = \begin{bmatrix} \cos^2(\theta) & \sin^2(\theta) & 0 & 0 & 0 & 2\cos(\theta)\sin(\theta) \\ \sin^2(\theta) & \cos^2(\theta) & 0 & 0 & 0 & -2\cos(\theta)\sin(\theta) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & 0 & 0 & \sin(\theta) & \cos(\theta) & 0 \\ -\cos(\theta)\sin(\theta) & \cos(\theta)\sin(\theta) & 0 & 0 & 0 & \cos^2(\theta) - \sin^2(\theta) \end{bmatrix}. \tag{2.18}$$

The inverse transpose of $\boldsymbol{T}(\theta)$ is explicitly given by the following expression

$$\boldsymbol{T}^{-\mathsf{T}}(\theta) = \begin{bmatrix} \cos^2(\theta) & \sin^2(\theta) & 0 & 0 & 0 & \cos(\theta)\sin(\theta) \\ \sin^2(\theta) & \cos^2(\theta) & 0 & 0 & 0 & -\cos(\theta)\sin(\theta) \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & \cos(\theta) & 0 \\ -2\cos(\theta)\sin(\theta) & 2\cos(\theta)\sin(\theta) & 0 & 0 & 0 & \cos^2(\theta) - \sin^2(\theta) \end{bmatrix}. \tag{2.19}$$

Given the following relation,

$$\boldsymbol{\sigma}_{123} = \boldsymbol{\mathcal{C}}_0 \boldsymbol{\varepsilon}_{123}. \tag{2.20}$$

The elasticity tensor $\boldsymbol{\mathcal{C}}(\theta)$ could be derived by substituting the stress and strain tensors in (2.20) by the ones in (2.17) which results in the following,

$$\boldsymbol{T}(\theta)\boldsymbol{\sigma}_{xyz} = \boldsymbol{\mathcal{C}}_0 \boldsymbol{T}^{-\mathsf{T}}(\theta)\boldsymbol{\varepsilon}_{xyz}. \tag{2.21}$$

Multiplying both sides by $\boldsymbol{T}^{-1}(\theta)$ yields,

$$\boldsymbol{\sigma}_{xyz} = \boldsymbol{T}^{-1}(\theta)\boldsymbol{\mathcal{C}}_0 \boldsymbol{T}^{-\mathsf{T}}(\theta)\boldsymbol{\varepsilon}_{xyz}. \tag{2.22}$$

Thus, the relation between the elasticity tensor in the global axes and in the one in the material axes reads,

$$\boldsymbol{\mathcal{C}}(\theta) = \boldsymbol{T}^{-1}(\theta)\boldsymbol{\mathcal{C}}_0 \boldsymbol{T}^{-\mathsf{T}}(\theta). \tag{2.23}$$

In the present thesis, we divide the domain into several different subdomains (layers or patches) and, as parameters to be optimized, we consider the fibre orientation angle in each single subdomain (layer or patch). Thus, the number of layers (or patches) $\mathtt{n_p}$ is the number of parameters characterizing the domain and is denoted by θ_i, such that $i = 1, \ldots, \mathtt{n_p}$. Each parameter θ_i ranges in a real interval $I_i \subset \mathbb{R}$ and describes the fiber orientation in a subdomain $\Omega_i \subset \Omega$ (note that the notation for the finite elements is Ω^e, $e = 1, \ldots, \mathtt{n_{el}}$, and typically many elements Ω^e are inside a subdomain Ω_i).

The $\mathtt{n_p}$ parameters are gathered in vector $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_{\mathtt{n_p}}]^{\mathsf{T}}$. Note that $\boldsymbol{\theta}$ ranges in the multidimensional parametric domain $I_{\boldsymbol{\theta}} = I_1 \times I_2 \times ... \times I_{\mathtt{n_p}} \subset \mathbb{R}^{\mathtt{n_p}}$.

Recalling (2.12), for every element e such that $\Omega^e \subset \Omega_i$, the element stiffness matrix reads,

$$\boldsymbol{K}^e(\theta_i) = \int_{\Omega^e} \boldsymbol{B}^{e\mathsf{T}} \boldsymbol{C}(\theta_i) \boldsymbol{B}^e d\Omega. \tag{2.24}$$

The parametric linear system of equations is derived using the parametric expression of the element stiffness matrix (2.24) in the assembly described in (2.16) resulting in

$$\boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{d}(\boldsymbol{\theta}) = \boldsymbol{f}. \tag{2.25}$$

It is worth noting that in this particular problem statement the force term $\boldsymbol{f}$ does not depend on the parameters.

2.4 Failure criteria

The design analysis of a composite laminate is performed by comparing the stresses due to the applied loads with an allowable strength of the material [70, 71]. To achieve such comparison, many failure criteria were proposed for different types of materials. It is also possible to quantify failure on the microscopic or the macroscopic level depending on the application. In the present work, we choose to take the phenomenological approach and quantify failure in a macroscopic way. In other words, our goal here is to have a measure of how good or bad is a design of a composite laminate based on the variation of fibre orientations rather than actually analyzing the mechanisms of failure. In a macroscopic approach, the strength of the composite laminate varies with the fibre orientation. The strength of a composite laminate along an arbitrary direction is related to the material's strength characteristics in a well defined material axes. In our case, where we assume the material is transversely isotropic, the material is characterized with five basic strength characteristics. Those characteristics are the longitudinal tensile and compressive strengths σ_{y}^{Lt} and σ_{y}^{Lc} respectively, the transverse tensile and compressive strengths σ_{y}^{Tt} and σ_{y}^{Tc} respectively, and the in-plane shear strength τ_{y}^{LT}. There are additional characteristics that arise when analyzing a composite laminate having less material symmetries.

Failure criteria could be classified based on the material type. For example, the Von Mises criterion is widely used for homogeneous isotropic material, such as steel and aluminium alloys, to detect yielding [71]. Those isotropic criteria are very well established and many works were dedicated to adapt those isotropic criteria to anisotropic ones [70]. Since a fibre reinforced composite laminate is not isotropic, many failure theories were proposed such as the maximum stress, maximum strain, the Tsai-Hill criterion, and the Tsai-Wu criterion. The differences among them have been discussed intensively in the literature and for deeper insight about different criteria, the reader is referred to the following literature [69–72]. In the following subsections we introduce some of those criteria

briefly and then present the Tsai-Wu criterion, that is used in our methodology, in detail.

2.4.1 Maximum stress, maximum strain, and Tsai-Hill criteria

The maximum stress criterion states that the material is considered at failure if any of the stress components in its principle directions exceeds the corresponding strength in the same direction [70]. Thus, for a transversely isotropic material to be safe, all the following inequalities have to hold,

$$\begin{aligned} -\sigma_{\mathrm{y}}^{Lc} <& \sigma_1 < \sigma_{\mathrm{y}}^{Lt} \\ -\sigma_{\mathrm{y}}^{Tc} <& \sigma_2, \sigma_3 < \sigma_{\mathrm{y}}^{Tt} \\ -\tau_{\mathrm{y}}^{LT} <& \sigma_5, \sigma_6 < \tau_{\mathrm{y}}^{LT} \\ -\tau_{\mathrm{y}}^{TV} <& \sigma_4 < \tau_{\mathrm{y}}^{TV}, \end{aligned} \tag{2.26}$$

where τ_{y}^{TV} is the out-of-plane (2-3 plane) yielding shear strength and the superscript letter "V" stands for the vertical direction. The stress state in the principle directions ($\boldsymbol{\sigma}_{123}$) relates to the applied stresses through Eq. (2.17) and, thus, the safe value of the applied stresses is affected by the change of the orientation of the material.

Similarly, according to the maximum strain criterion, the material is considered at failure if any of the strains in the principle directions exceeds its corresponding ultimate strain in the same direction [70]. Thus for a transversely isotropic material, the material is considered safe if and only if the following inequalities hold,

$$\begin{aligned} -\varepsilon_{\mathrm{y}}^{Lc} <& \varepsilon_1 < \varepsilon_{\mathrm{y}}^{Lt} \\ -\varepsilon_{\mathrm{y}}^{Tc} <& \varepsilon_2, \varepsilon_3 < \varepsilon_{\mathrm{y}}^{Tt} \\ -\varepsilon_{\mathrm{y}}^{LT} <& \varepsilon_5, \varepsilon_6 < \varepsilon_{\mathrm{y}}^{LT} \\ -\varepsilon_{\mathrm{y}}^{TV} <& \varepsilon_4 < \varepsilon_{\mathrm{y}}^{TV}, \end{aligned} \tag{2.27}$$

where $\varepsilon_{\mathrm{y}}^{Lt}, \varepsilon_{\mathrm{y}}^{Lc}, \varepsilon_{\mathrm{y}}^{Tt}, \varepsilon_{\mathrm{y}}^{Tc}, \varepsilon_{\mathrm{y}}^{LT}, \varepsilon_{\mathrm{y}}^{TV}$ are the ultimate longitudinal tensile, longitudinal compressive, transverse tensile, transverse compressive, in-plane shear, out-of-plane shear strains respectively. The strains are obtained first by transforming the stresses as shown in equation Eq. (2.17). Then finding the corresponding strains by applying the inverse of the elasticity matrix (compliance matrix) to the stresses. This theory accounts for some interaction between stresses due to the Poisson's ratio effect.

Following the work by Hill extending the Von Mises criterion for anisotropic ductile metal, Azzi and Tsai adapted Hill's criterion to orthotropic composite materials. In the case of a 3D stress state, the Azzi-Tsai-Hill criterion states that the material is considered

safe if the following inequality holds,

$$\frac{\sigma_1^2 - \sigma_1\sigma_2 - \sigma_1\sigma_3}{\mathbb{f}_1^2} + \frac{\sigma_2^2 + \sigma_3^2 - \sigma_2\sigma_3}{\mathbb{f}_2^2} + \frac{\tau_4^2}{\mathbb{f}_4^2} + \frac{\tau_5^2 + \tau_6^2}{\mathbb{f}_6^2} < 1, \tag{2.28}$$

where the strength parameter $\mathbb{f}$ is particularized to the material basic strength characteristics introduced in Section 2.4.2 depending on the stress state. Thus,

$$\mathbb{f}_1 = \begin{cases} \sigma_{\mathrm{y}}^{Lt} & \text{when} \quad \sigma_1 > 0 \\ \sigma_{\mathrm{y}}^{Lc} & \text{when} \quad \sigma_1 < 0 \end{cases} \tag{2.29a}$$

$$\mathbb{f}_2 = \begin{cases} \sigma_{\mathrm{y}}^{Tt} & \text{when} \quad \sigma_2 > 0 \\ \sigma_{\mathrm{y}}^{Tc} & \text{when} \quad \sigma_2 < 0 \end{cases} \tag{2.29b}$$

$$\mathbb{f}_4 = |\tau_{\mathrm{y}}^{TV}| \tag{2.29c}$$

$$\mathbb{f}_6 = |\tau_{\mathrm{y}}^{LT}|, \tag{2.29d}$$

where for transversely isotropic materials, $\mathbb{f}_2 \cong \mathbb{f}_3$ and $\mathbb{f}_5 \cong \mathbb{f}_6$.

The criterion showed superiority over the maximum stress and maximum strain theories as it is expressed with a single criterion instead of many sub-criteria; and it also has a good fit with the experimental data. One of the main disadvantages is that it cannot differentiate between tensile or compressive stresses given its quadratic nature as shown in equation Eq. (2.28). This means that the stress state has to be specified and accordingly the proper strengths terms are chosen as in Eq. (2.29).
Finally we resort to the Tsai-Wu criterion, that is widely used nowadays, for three main reasons. The first reason is because it matches best the experimental data out of all the presented criteria in the current section, and the second reason is that the criterion takes into account tensile and compressive stress and multi-axial stress states accounting for stress interaction, and finally it is expressed using one single scalar function that is easy to implement [70, 71, 73].

2.4.2 Tsai-Wu criterion

Tsai-Wu criterion is a general anisotropic theory that is an extension of the Azzi-Tsai-Hill criterion. The criterion is intended to be a descriptive tool for us to have a general notion of the load-bearing capacity of structures for design purposes. The failure criterion proposed by Tsai and Wu [73] is based on the scalar failure index $\mathcal{I}_{\mathtt{f}}$ defined as function of the stress $\boldsymbol{\sigma}$ as follows:

$$\mathcal{I}_{\mathtt{f}}(\boldsymbol{\sigma}) = \boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{\mathcal{F}}\boldsymbol{\sigma} + \boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{F}. \tag{2.30}$$

In particular, the criterion states that the material at point $\boldsymbol{x}$ is not at failure if and only if $\mathcal{I}_{\mathtt{f}}(\boldsymbol{\sigma}) \leq 1$. The dependency of the failure index on the position $\boldsymbol{x}$ is through the stress state $\boldsymbol{\sigma}(\boldsymbol{x})$. The strength tensors $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{F}$ are fourth and second order tensors respectively (expressed as a matrix and a vector, when using the engineering Voigt's notation).

There are several features of this criterion that make it a suitable choice over other criteria [73]:

(1) The criterion is a scalar equation. It accounts for interactions among all stresses, unlike the maximum stress and strain theories where interactions are not admissible.
(2) Since the strength components are represented using second and fourth order tensors, their transformations are well established and very similar to the one used for the elasticity tensor. Moreover the tensors are analogous to the elasticity tensor in terms of symmetry properties and number of non zero and independent components.
(3) The anisotropy and multidimensional spaces don't cause any difficulty.
(4) There is flexibility in the representation of the failure criterion in terms of transformations. It could be represented either in material axes or in global axes.
(5) The criterion is automatically invariant which makes it valid for all coordinate systems.
(6) A stability condition is incorporated to ensure that the failure envelope is not open ended.

$$F_{mm}F_{nn} - F_{mn}^2 \geq 1, \tag{2.31}$$

where the indices m and n are not summation indices and they run from 1 to 6 to represent different components of the fourth order strength tensor.

The full strength tensors have the following form for an anisotropic material,

$$\boldsymbol{\mathcal{F}} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} & F_{15} & F_{16} \\ & F_{22} & F_{23} & F_{24} & F_{25} & F_{26} \\ & & F_{33} & F_{34} & F_{35} & F_{36} \\ & Sym. & & F_{44} & F_{45} & F_{46} \\ & & & & F_{55} & F_{56} \\ & & & & & F_{66} \end{bmatrix} \tag{2.32}$$

$$\boldsymbol{F} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix}. \tag{2.33}$$

The strength tensors are treated like the elasticity tensor in terms of the reduction of the number of independent components due to the stress, strain, and material symmetries. The number of independent components for anisotropic material is 21 and 6 for $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{F}$ respectively. For a transversely isotropic material, the number of independent components reduces to 5 and 3 for $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{F}$ respectively. Thus, the forms of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{F}$ in the material axes, denoted by $\boldsymbol{\mathcal{F}}_0$ and $\boldsymbol{F}_0$, are

$$\boldsymbol{\mathcal{F}}_0 = \begin{bmatrix} F_{11} & F_{12} & F_{12} & 0 & 0 & 0 \\ & F_{22} & F_{23} & 0 & 0 & 0 \\ & & F_{22} & 0 & 0 & 0 \\ & \text{Sym.} & & 2(F_{22}-F_{23}) & 0 & 0 \\ & & & & \dfrac{1}{(\tau_{\mathtt{y}}^{TV})^2} & 0 \\ & & & & & \dfrac{1}{(\tau_{\mathtt{y}}^{TV})^2} \end{bmatrix} \tag{2.34}$$

$$\boldsymbol{F}_0 = \begin{bmatrix} F_1 \\ F_2 \\ F_2 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.35}$$

Remark 2.1. Tensors $\boldsymbol{\mathcal{F}}_0$ and $\boldsymbol{F}_0$ are material characteristics that have to be determined in the laboratory. This is achieved by applying uni-axial and bi-axial stresses in tension and compression and measuring the failure strengths [73].

$$F_{11} = \frac{1}{\sigma_{\mathtt{y}}^{Lt}\sigma_{\mathtt{y}}^{Lc}}, \quad F_1 = \frac{1}{\sigma_{\mathtt{y}}^{Lt}} - \frac{1}{\sigma_{\mathtt{y}}^{Lc}}, \tag{2.36}$$

$$F_{22} = \frac{1}{\sigma_{\mathtt{y}}^{Tt}\sigma_{\mathtt{y}}^{Tc}}, \quad F_2 = \frac{1}{\sigma_{\mathtt{y}}^{Tt}} - \frac{1}{\sigma_{\mathtt{y}}^{Tc}}, \tag{2.37}$$

$$F_{12} = -\frac{1}{2}\sqrt{F_{11}F_{22}}, \quad F_{23} = -\frac{1}{2}\sqrt{F_{22}F_{33}}, \tag{2.38}$$

where, $\sigma_{\mathtt{y}}^{Lt}$, $\sigma_{\mathtt{y}}^{Lc}$, $\sigma_{\mathtt{y}}^{Tt}$, $\sigma_{\mathtt{y}}^{Tc}$ and $\tau_{\mathtt{y}}^{TV}$ are the longitudinal tensile, longitudinal compressive, transverse tensile, transverse compressive and shear yielding strength of the material respectively, and the subscript "$\mathtt{y}$" stands for *yield* state of the strength.

In order to apply the Tsai-Wu criterion, the stress states and the strength tensors have to be evaluated in the same coordinate system. Therefore, it is necessary to either represent the stresses in the material axes (principle directions) or transform and represent the strength tensors in the global axes [73]. Conceptually both ways give the same results represented in different axes and one could choose one way over the other for computational simplicity later on in the PGD framework.

Equation (2.30) is expanded as follows when the computed stresses are transformed to the material axes for transversely isotropic material with 2-3 as the plane of isotropy;

$$\begin{aligned}\mathcal{I}_{\mathtt{f}} =& F_1\sigma_1 + F_2(\sigma_2+\sigma_3) + ... \\ & F_{11}\sigma_1^2 + 2F_{12}(\sigma_1\sigma_2+\sigma_1\sigma_3) + 2F_{23}(\sigma_2\sigma_3-\sigma_4^2) + ... \\ & F_{22}(\sigma_2^2+\sigma_3^2+2\sigma_4^2) + F_{66}(\sigma_5^2+\sigma_6^2).\end{aligned} \tag{2.39}$$

The other way is when the computed stresses are left unchanged and the strength tensors are transformed. We denote the components of the transformed strength tensors $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{F}$ with a prime "$\prime$", and the vector of computed stresses reads $\boldsymbol{\sigma} = [\sigma_{xx}\ \sigma_{yy}\ \sigma_{zz}\ \sigma_{yz}\ \sigma_{xz}\ \sigma_{xy}]^{\mathsf{T}}$. In that case, Eq. (2.30) is expanded as follows for transversely isotropic material with plane 2-3 as plane of isotropy [73],

$$\begin{aligned}\mathcal{I}_{\mathtt{f}} =& F'_1\sigma_{xx} + F'_2\sigma_{yy} + F'_3\sigma_{zz} + F'_6\sigma_{xy} + F'_{11}\sigma_{xx}^2 + F'_{22}\sigma_{yy}^2 + F'_{33}\sigma_{zz}^2 + ... \\ & F'_{44}\sigma_{yz}^2 + F'_{55}\sigma_{xz}^2 + F'_{66}\sigma_{xy}^2 + 2F'_{12}\sigma_{xx}\sigma_{yy} + 2F'_{23}\sigma_{yy}\sigma_{zz} + ... \\ & 2F'_{31}\sigma_{zz}\sigma_{xx} + 2F'_{16}\sigma_{xx}\sigma_{xy} + 2F'_{26}\sigma_{yy}\sigma_{xy} + 2F'_{36}\sigma_{zz}\sigma_{xy} + ... \\ & 2F'_{45}\sigma_{yz}\sigma_{xz}.\end{aligned} \tag{2.40}$$

The Tsai-Wu criterion in (2.30) is non-homogeneous, meaning that it has a quadratic term and a linear term, where the latter takes into account the internal stresses that differentiate between tensile and compressive stress states [73]. It was demonstrated in [30] that the Tsai-Wu criterion could be load dependent. This means that for applied loads under a certain threshold, the criterion in (2.30) is dominated by the linear term, and thus, leading to inaccurate counter-intuitive optimization results. To alleviate this problem, the failure criterion may be alternatively expressed in terms of a scalar load multiplier λ. The load multiplier (or safety factor) λ produces a stress state $\overline{\boldsymbol{\sigma}} = \lambda\boldsymbol{\sigma}$, such that, the applied stress scales by a factor λ to match the yielding state of the material at the onset of failure [30]. Consequently, the goal now is to find the best fibre orientation that maximizes the safety factor λ, and therefore, design the laminate based on the maximum load-bearing capacity of the structure just before failure. Thus, the corresponding failure index reads,

$$\mathcal{I}_{\mathtt{f}}\left(\overline{\boldsymbol{\sigma}}\right) = \mathcal{I}_{\mathtt{f}}\left(\lambda\boldsymbol{\sigma}\right) = \lambda^2\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{\mathcal{F}}\boldsymbol{\sigma} + \lambda\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{F}. \tag{2.41}$$

At each point $\boldsymbol{x}$, the critical value of λ corresponds to the onset of failure $\mathcal{I}_{\mathtt{f}}\left(\overline{\boldsymbol{\sigma}}\right) = 1$. This results in a polynomial equation of the second degree for λ where its explicit solution is available. Assuming that $\boldsymbol{\mathcal{F}}$ is symmetric positive definite, $\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{\mathcal{F}}\boldsymbol{\sigma} \geq 0$, there is a unique positive root of the equation $\mathcal{I}_{\mathtt{f}}\left(\overline{\boldsymbol{\sigma}}\right) = 1$. The smallest positive root is the safety factor, denoted by λ_s and reads,

$$\lambda_s = \frac{1}{2\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}}\left(\sqrt{(\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{F})^2 + 4\,\boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}} - \boldsymbol{\sigma}^{\mathsf{T}}\boldsymbol{F}\right). \tag{2.42}$$

Note that the safety factor depends on the choice of the parameters $\boldsymbol{\theta}$ and the point $\boldsymbol{x}$ where the stress is evaluated; and thus the notation $\lambda_s(\boldsymbol{x}, \boldsymbol{\theta})$ is adopted in the following.

To emphasize the dependence of the failure criterion in (2.30) on the fibre orientation (our parameters), we explicitly express the strength tensors in (2.34) and (2.35) with respect to the global axes. Thus, in an arbitrary point $\boldsymbol{x} \in \Omega_i$ where the fiber orientation angle is θ_i, or in any element $\Omega^e \subset \Omega_i$, the expressions of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{F}$ with respect to the global axes are obtained using the transformation matrices, introduced in (2.18) and (2.19), namely

$$\begin{aligned} \boldsymbol{\mathcal{F}}(\theta_i) &= \boldsymbol{T}^{\mathsf{T}}(\theta_i)\boldsymbol{\mathcal{F}}_0\boldsymbol{T}(\theta_i) \\ \boldsymbol{F}(\theta_i) &= \boldsymbol{T}^{\mathsf{T}}(\theta_i)\boldsymbol{F}_0. \end{aligned} \tag{2.43}$$

It is worth noting that the dependence of $\mathcal{I}_{\mathtt{f}}(\boldsymbol{\sigma})$ on $\boldsymbol{\theta}$ does not only come from $\boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{\theta})$ but also from $\boldsymbol{\mathcal{F}}(\theta_i)$ and $\boldsymbol{F}(\theta_i)$ where they depend only on the local θ_i because they are material properties. Marking explicitly the parametric dependence, for $\boldsymbol{x} \in \Omega_i$, equation (2.30) is rewritten as:

$$\mathcal{I}_{\mathtt{f}}\big(\boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{\theta})\big) = \boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{\mathcal{F}}(\theta_i)\boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{\sigma}(\boldsymbol{x}, \boldsymbol{\theta})^{\mathsf{T}}\boldsymbol{F}(\theta_i). \tag{2.44}$$

Equation (2.44) and the smallest positive root of Equation (2.42) are the objective functions of our optimization problem. In the following section we introduce the optimization problem. We also briefly introduce different types of algorithms that could be used to solve the problem.

2.5 Optimization problem

Optimization is the process of finding the best design variables that minimizes or maximizes a particular design objective of a given problem. Following [30], two alternative optimization problems are considered to find the optimal values of the parameters $\boldsymbol{\theta}$.

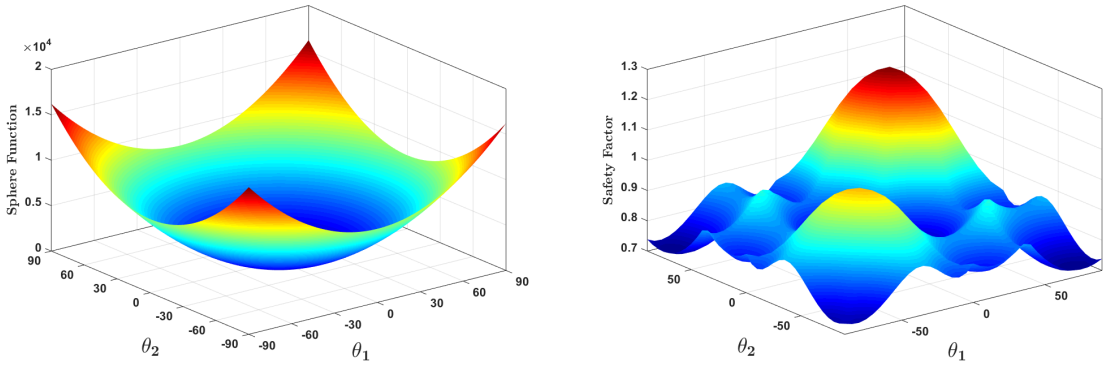
The first choice is to find $\boldsymbol{\theta}$ that minimizes the maximum value of $\mathcal{I}_{\mathtt{f}}\big(\boldsymbol{\sigma}(\boldsymbol{x},\boldsymbol{\theta})\big)$ evaluated at all points $\boldsymbol{x}$ in Ω. Thus, the optimization problem reads,

$$\boldsymbol{\theta}_{\mathtt{f}}^{\mathtt{Opt}} = \arg\min_{\boldsymbol{\theta}} \max_{\boldsymbol{x}} \mathcal{I}_{\mathtt{f}}\big(\boldsymbol{\sigma}(\boldsymbol{x},\boldsymbol{\theta})\big), \tag{2.45}$$

where superscript "Opt" is used to indicate the optimal choice and subscript "f" is a label to indicate that the objective function is based on the failure index. Alternatively, the second choice is to find $\boldsymbol{\theta}$ that maximizes the minimum value of $\lambda_s(\boldsymbol{x},\boldsymbol{\theta})$ evaluated at all points $\boldsymbol{x}$ in Ω. The corresponding optimal choice (labeled by subscript "s" for safety) is

$$\boldsymbol{\theta}_{\mathtt{s}}^{\mathtt{Opt}} = \arg\max_{\boldsymbol{\theta}} \min_{\boldsymbol{x}} \lambda_s(\boldsymbol{x},\boldsymbol{\theta}). \tag{2.46}$$

The objective functions in Equations 2.42 and 2.44 are not necessarily smooth and they are non convex-concave which might lead to being stuck in local minima/maxima [30]. In Fig. 2.5b is shown an example of a minimum safety factor function to be maximized. It is demonstrated that the function has many local minima and local maxima.



(a) Convex sphere function

(b) Non-convex function

Figure 2.5: Convex and non-convex functions example

Accordingly, the choice of the optimization algorithm is of paramount importance to avoid false optimal solutions. The optimization algorithm is an iterative procedure applied to the problem with the aim of reaching a satisfactory solution. The classification of such algorithms is discussed extensively in the literature [74]. Generally the optimization algorithms are classified into deterministic and stochastic algorithms. The deterministic algorithms follow well defined paths to find the optimal solution of a certain problem. In other words, if we provide the algorithm with the same starting point, it will take the same search path and reach the same optimal solution. One of the most famous and widely used examples of such algorithms is gradient-based methods. On the other hand,

the stochastic algorithms follow random different paths, each time the code is running, in an attempt to find the optimal solution. The Genetic Algorithm (GA) is a good example. There are further subdivisions of the deterministic and stochastic algorithms that could be found in detail in [74] and the references therein. In the current thesis we use both an evolutionary method alone and we use a hybrid of a gradient-based method and an evolutionary method to optimize the fibre orientations. In the hybrid approach, we aim to run first the evolutionary method without a large accuracy benefiting from its global search character, then use the gradient-based algorithm to seek the solution locally in a fast manner.

Gradient-based methods use gradient information to decide the direction of the search for the minimum or maximum of a function. The use of such methods in convex problems (as in Fig. 2.5a) is very appropriate and the convergence to the optimal solution is very fast. However, in non-convex problems as shown in Fig. 2.5b, the gradient-based algorithm tends to be stuck in a local minimum (or maximum) if the initial guess is far from the global optimal solution. Moreover, the evaluation of the gradients (and sometimes the Hessian) is required in some methods which could be a difficult task if they depend on the design variables implicitly. Fortunately in our case, the PGD framework provides solutions depending explicitly on the design variables and, consequently, it is very easy to compute the gradient and the Hessian.

On the other hand, the global method we use is an evolutionary method, namely the Genetic Algorithm (GA). GA is an algorithm that tries to mimic the natural evolution in biology that is based on Darwin's theory of natural selection and the survival of the fittest. GA has two main advantages: it deals with a variety of complex optimization problems yielding a global solution and parallelism [74]. The main steps in a GA are as follows:

- The selection criterion or the fitness function definition;
- Initializing a population;
- Evaluating the fitness of the whole population;
- Create new offspring by performing crossover, mutation, and selection of the fittest.
- Advance the algorithm in the same manner until a stopping condition is met;
- Finally, decode the results to analyse the solution.

The fitness function quantifies how good are the elements of the population to be a candidate optimal solution. Initially the elements of the population (possible solutions) are encoded in binary arrays and are randomly selected. Next, the fitness function is evaluated for all candidate solutions (our initial population). The iteration where we evaluate the fitness function is called a *generation*. The process is repeated until we reach a predefined threshold, normally the maximum number of generations.

Algorithm 2.1: Pseudo-code of the genetic algorithm

Input : Number of generations, population size, crossover probability, mutation probability

Output: Optimal global solution

(1) Choose the objective function $f(x)$ for the problem

(2) Encode the solution into binary arrays

(3) Define a fitness function F such that $F \propto f(x)$

(4) Initialize a population of possible solutions randomly

(5) $\mathtt{N_g} \leftarrow 0$

while $N_g <$ *Number of generations* **do**

(5.1) Generate new solution by the crossover and mutation operators:

if $p_c >$*rand* **then**

- Apply crossover

end

if $p_m >$*rand* **then**

- Apply mutation

end

(5.2) Select the best solution in terms of fitness

(5.3) $\mathtt{N_g} \leftarrow \mathtt{N_g} + 1$

(6) Decode and analyse the results

In the following chapter we present the PGD framework in detail that is used to solve the problem presented in the current chapter.

Chapter 3

PROPER GENERALIZED DECOMPOSITION (PGD) FRAMEWORK

3.1 Preliminaries

This section briefly describes the Proper Generalized Decomposition (PGD) as a tool to obtain a parametric solution of the problem described in Section 2.3 that depends explicitly on the fibre orientation. This explicit parametric solution, also denoted as *computational vademecum*, allows expressing in a compact form the solutions corresponding to all possible values of parameters $\boldsymbol{\theta}$. In a nutshell, the main concepts behind the PGD approach are summarized in three steps as follows [9, 49]:

- First, the parameters are taken as extra coordinates, stating the problem in a multidimensional framework; this means finding an approximation to $\boldsymbol{d}(\boldsymbol{\theta})$ in $\mathbb{R}^{n_d} \times \boldsymbol{I}_{\boldsymbol{\theta}}$, i.e. the solution of (2.25). Consequently, the multidimensional character of the problem drastically increases its computational complexity (the number of degrees of freedom is the product of the number of degrees of freedom in each parametric dimension). This cannot be solved with a standard discretization method in the multidimensional domain.
- Second, in order to reduce the computational complexity, the solution is sought in a separable format. This means that the solution is written as a sum of products

of sectional functions each depending only on one of the parameters; each term is referred to as a rank-one term. Thus, the actual number of degrees of freedom reduces to the sum of the number of degrees of freedom in each parametric dimension.

- Third, the algorithm to solve this problem is based on a greedy strategy (computing one rank-one term at a time) and an alternating directions method to solve the nonlinear rank-one problems.

The computational vademecum is typically computed in an *offline* phase that may take important computational resources. The interesting aspect of the PGD is that once the explicit parametric solution is available, exploring the parametric space (e.g. for an optimization problem) is a simple postprocess, which is extremely fast, and it can be conducted *online* in real time. In the following we show the *encapsulated* concept, presented in [50, 51], where a set of algebraic tools operate with multidimensional tensors in a separable format. Subsequently, we present the process of separation of input for the encapsulated PGD, and finally we introduce the post-process steps to compute the failure index and ultimately solve the optimization problem. The chapter is organized as shown is Fig. 3.1.

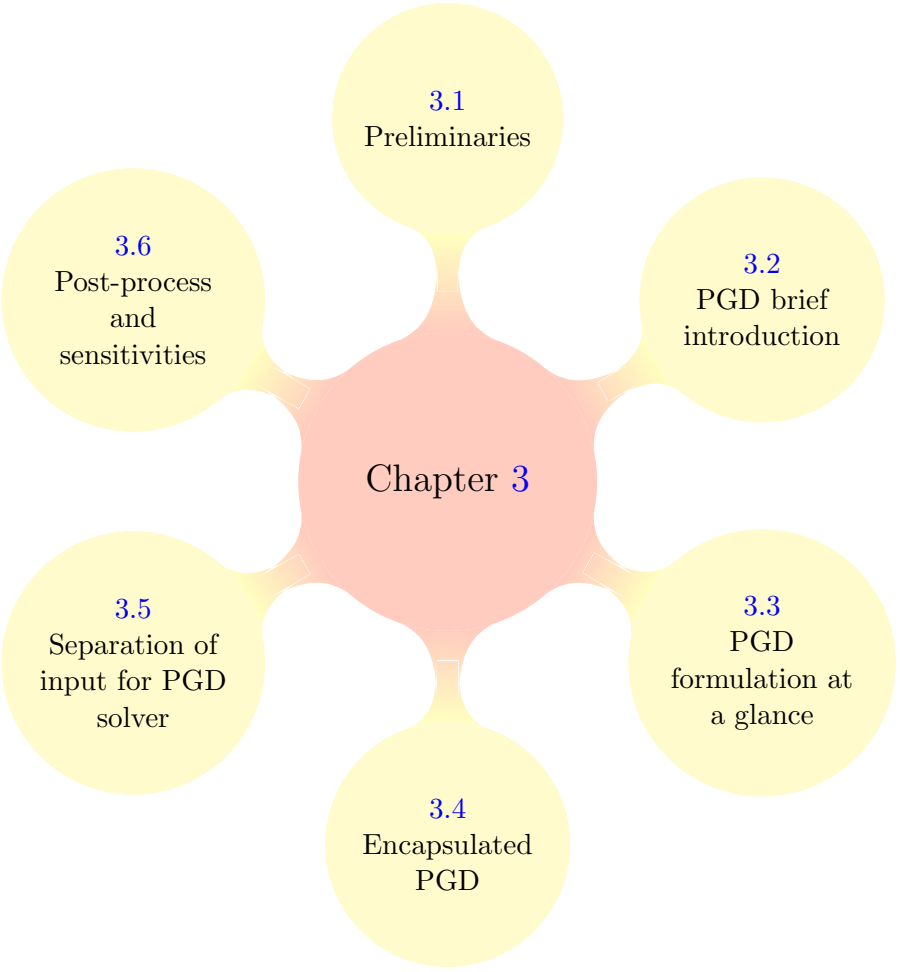


Figure 3.1: Structure of Chapter 3

3.2 PGD brief introduction

The PGD is a member of the Model Order Reduction (MOR) family. In the present work, PGD is used to obtain a computational vademecum accounting for parameters representing the fibre orientations in a composite laminate as defined in Section 2.3. In order to do so, we need to consider our parameters as new independent variables (extra-coordinates) in the problem. In a space of dimension $\mathtt{N_d}$, we could concretely say that the solution for a general field $u(\vartheta_1, \vartheta_2, ..., \vartheta_{\mathtt{N_d}})$ is sought in a multidimensional space $\mathcal{D} = \mathcal{S}_1 \times \mathcal{S}_2 \times ... \times \mathcal{S}_{\mathtt{N_d}} \subset \mathbb{R}^{\mathtt{N_d}}$ where each coordinate $\vartheta_i \in \mathcal{S}_i$ with $i = 1, \ldots, \mathtt{N_d}$. The coordinate ϑ_i could be a spatial, time, or material coordinate. In this way we could solve the problem only once in the so-called *offline phase* and then perform extremely fast computations to obtain the solution for a given value of the set of parameters in the so-called *online phase*. This is particularly useful in a many-query problem, like optimization or inverse problems, where the direct model has to be evaluated several times for different values of the set of parameters. However, the computational complexity explodes with the increase of independent variables which is referred to as *curse of dimensionality* [49].

Solving this multidimensional problem using a standard mesh based discretization technique would lead to an exponential increase in the computational complexity. In other words, solving a model with $\mathtt{N_d}$ dimensions, where each space coordinate is discretized using $\mathcal{M}$ nodes, the resulting total number of degrees of freedom is $\mathcal{M}^{\mathtt{N_d}}$. For example, with $\mathcal{M} = 1000$ and $\mathtt{N_d} = 30$, the resulting numerical complexity is 10^{90} which is an astronomical number [49]. One way to circumvent the cruse of dimensionality is to represent the solution in a separable manner. Separated representations is a very well established technique inspired from Fourier and introduced by Ladeveze in numerous works [49]. Thus the PGD framework approximates the field u in a separated form as follows,

$$u(\vartheta_1, \vartheta_2, ..., \vartheta_{\mathtt{N_d}}) \approx \sum_{m=1}^{n} \mathcal{V}_1^m(\vartheta_1) \cdot \mathcal{V}_2^m(\vartheta_2) \cdot ... \cdot \mathcal{V}_{\mathtt{N_d}}^m(\vartheta_{\mathtt{N_d}}) = \sum_{m=1}^{n} \prod_{j=1}^{\mathtt{N_d}} \mathcal{V}_j^m(\vartheta_j). \tag{3.1}$$

The PGD approximation is, therefore, a sum of n functional products of $\mathtt{N_d}$ functions $(\mathcal{V}_j^m(\vartheta_j))$ that are unknown *a priori* [49]. As a result of this approximation, the total number of degrees of freedom decreases from $\mathcal{M}^{\mathtt{N_d}}$ to $n \cdot \mathcal{M} \cdot \mathtt{N_d}$, that is, a linear increase of the computational complexity with the increase of number of independent variables in the problem rather than an exponential increase. The PGD approximation is constructed by successive enrichments using a greedy strategy and each functional product is determined

sequentially. This means that we compute the term $m = 1$, then $m = 2$, and so on until we reach a certain predefined threshold. Note that, the number of enrichment terms n is not known a priori. In theory, to obtain an exact solution, the number of enrichment terms should be infinite. However this is impractical and one chooses a certain level of accuracy where the enrichment of the solution stops and a finite number of terms is accepted. Let's assume we are computing the functions at step $\tilde{n}$. The solution is already computed in the previous steps, that is $m < \tilde{n}$ and is enriched by computing and adding the current step. Each function of the current step is computed sequentially. This leads to a series of low-dimensional non-linear problems to be solved, at each step m, using an alternated direction fixed point strategy [49], hence reducing the order of the model. The non-linear nature of each problem at each enrichment step arises from the fact that we need to solve for the functions $\mathcal{V}_j^m(\vartheta_j)$ that are multiplying each other.

One of the main advantages of PGD is that it provides a generalized solution which allows us to reconstruct the solution extremely fast by particularizing it to the desired set of parameters. Another advantage is that the dependence on the parameters is explicit which facilitates the computation of the sensitivities for the optimization problem. There are also numerous applications where PGD fits very well such as in geophysics [75], materials [56], bio-mechanics [76], and virtual surgeries [49] and many more. In the following section we show briefly how the PGD is adapted to 3D linear elasticity where we do not separate the space and only separate the parameters.

3.3 PGD formulation at a glance

Following the work of [49, 63, 76], in the current section the PGD formulation is briefly explained for the problem presented in Chapter 2. Therefore, the unknown we are solving for, in 3D, is the displacement field $\boldsymbol{u}$ presented in problem (2.1). In our problem, the aim is to obtain a generalized solution parameterized with the fibre orientations, as coordinates, instead of solving the model for discrete values of the orientations. This means we introduce the fibre orientations as independent variables along with the spatial coordinates into the problem formulation. In this case, the displacement field is generalized from $\boldsymbol{u}(\boldsymbol{x})$ to $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_{\mathtt{n_p}}]^\mathsf{T} \in I_{\boldsymbol{\theta}}$ denotes a vector holding $\mathtt{n_p}$ parameters. Since we are interested in a parametric PGD in the fibre orientations, there is no need to separate our spatial space. Thus, the output we are aiming for is a generalized solution that is explicit in the parameters, corresponding to different layers (or patches), and it is reconstructed very quickly when particularized to a set of parameters to give the mechanical response of the system (i.e. displacements).
Let us start from the weak form of our problem from Eq. (2.5).

Given a parametric space $I_{\boldsymbol{\theta}} = I_1 \times I_2 \times \cdots \times I_{\mathtt{n_p}}$, we solve for $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{\theta}) \in U \bigotimes_{j=1}^{\mathtt{n_p}} L^2(I_j)$,

$$\int_{\Omega} (\boldsymbol{\nabla}_{\mathtt{s}} \boldsymbol{w}^*)^{\mathsf{T}} \boldsymbol{C} \boldsymbol{\nabla}_{\mathtt{s}} \boldsymbol{u} \, d\Omega = \int_{\Gamma_N} \boldsymbol{w}^{*\mathsf{T}} \boldsymbol{t} d\Gamma + \int_{\Omega} \boldsymbol{w}^{*\mathsf{T}} \boldsymbol{b} d\Omega, \tag{3.2}$$

where $\boldsymbol{w^*}(\boldsymbol{x}, \boldsymbol{\theta}) \in U_0 \bigotimes_{j=1}^{\mathtt{n_p}} L^2(I_j)$ is the test function for the PGD formulation chosen in an appropriate space. The main steps of the formulation are introduced next with two parameters only to ease the notation, however, it is valid for any number of parameters $\mathtt{n_p}$ and poses no additional difficulty.

We are seeking a solution having the following form,

$$\boldsymbol{u}(\boldsymbol{x}, \theta_1, \theta_2) \approx \boldsymbol{u}^n(\boldsymbol{x}, \theta_1, \theta_2) = \sum_{m=1}^{n} \mathcal{X}^m(\boldsymbol{x}) \cdot \Theta_1^m(\theta_1) \cdot \Theta_2^m(\theta_2), \tag{3.3}$$

where $\mathcal{X}^m(\boldsymbol{x})$ is the spatial function with $\boldsymbol{x} \in \Omega \subset \mathbb{R}^3$ and $\Theta_j^m(\theta_j)$ represents the parametric function for the jth parameter $\theta_j \in I_j \subset \mathbb{R}$. Our goal now is to obtain an accurate enough approximation of the solution shown in Eq. (3.3). As mentioned previously, the greedy strategy computes each enrichment term sequentially.

Let us assume we would like to compute the solution at the enrichment step $\tilde{n}$ and that we already know the solution at enrichment step $\tilde{n} - 1$ such that

$$\boldsymbol{u}^{\tilde{n}}(\boldsymbol{x}, \theta_1, \theta_2) = \boldsymbol{u}^{\tilde{n}-1}(\boldsymbol{x}, \theta_1, \theta_2) + \mathcal{X}^{\tilde{n}}(\boldsymbol{x}) \cdot \Theta_1^{\tilde{n}}(\theta_1) \cdot \Theta_2^{\tilde{n}}(\theta_2). \tag{3.4}$$

For the sake of notation simplicity, we will drop the dependence on the current step $\tilde{n}$ and we replace the functional product $\mathcal{X}^{\tilde{n}}(\boldsymbol{x}) \cdot \Theta_1^{\tilde{n}}(\theta_1) \cdot \Theta_2^{\tilde{n}}(\theta_2)$ by $\mathcal{R}(\boldsymbol{x}) \cdot S_1(\theta_1) \cdot S_2(\theta_2)$. The solution at step $\tilde{n}$ could be rewritten as

$$\boldsymbol{u}^{\tilde{n}}(\boldsymbol{x}, \theta_1, \theta_2) = \boldsymbol{u}^{\tilde{n}-1}(\boldsymbol{x}, \theta_1, \theta_2) + \mathcal{R}(\boldsymbol{x}) \cdot S_1(\theta_1) \cdot S_2(\theta_2). \tag{3.5}$$

According to [49], the simplest choice of the test functions $\boldsymbol{w}^*$ could be taken as

$$\boldsymbol{w}^* = \mathcal{R}^*(\boldsymbol{x}) \cdot S_1(\theta_1) \cdot S_2(\theta_2) + \mathcal{R}(\boldsymbol{x}) \cdot S_1^*(\theta_1) \cdot S_2(\theta_2) + \mathcal{R}(\boldsymbol{x}) \cdot S_1(\theta_1) \cdot S_2^*(\theta_2), \tag{3.6}$$

where we denote the unknown terms that we want to compute by an asterisk "*". Incorporating the test functions in Eq. (3.6) and the trial functions in Eq. (3.5) in the weak form in Eq. (3.2), it results in a non-linear problem that is solved using an alternated directions fixed point algorithm. This algorithm is chosen for its robustness and ease of implementation [49]. In each iteration, three steps are repeatedly performed until we reach convergence:

(1) Computing $\mathcal{R}^*(\boldsymbol{x})$ assuming that the terms $S_1(\theta_1)$ and $S_2(\theta_2)$ are already known from the previous iteration. In this case the test function reduces to $\mathcal{R}^*(\boldsymbol{x}) \cdot S_1(\theta_1) \cdot S_2(\theta_2)$ since the only unknown is $\mathcal{R}^*(\boldsymbol{x})$.

(2) Using the just computed $\mathcal{R}(\boldsymbol{x})$ and $S_2(\theta_2)$ computed from the previous iteration to compute $S_1(\theta_1)$ with the test function as $\mathcal{R}(\boldsymbol{x}) \cdot S_1^*(\theta_1) \cdot S_2(\theta_2)$.

(3) Update $S_2^*(\theta_2)$ using the just-updated quantities $\mathcal{R}(\boldsymbol{x})$ and $S_1(\theta_1)$. In this case the test function becomes $\boldsymbol{w}^* = \mathcal{R}(\boldsymbol{x}) \cdot S_1(\theta_1) \cdot S_2^*(\theta_2)$.

Note that for the initial enrichment step $n = 1$, an arbitrary guess is used. The obtained converged functions are the functional product for the enrichment step $\tilde{n}$. The enrichment procedure is repeated until the greedy algorithm tolerance is reached.

As can be seen, the PGD is composed of a hierarchical two-loop structure. The PGD enrichment is sought by the outer-loop and the inner-loop solves, in an iterative way, for the functional product that expresses the spatial and parametric problems. Both loops are controlled using predefined tolerances to attain an accurate result. In order to control the loops, we define the amplitude of the $\tilde{n}$th mode as follows

$$A^{\tilde{n}} := \|\mathcal{X}^{\tilde{n}}\| \prod_{j=1}^{\mathtt{n_p}} \|\Theta_j^{\tilde{n}}\|, \tag{3.7}$$

where $\|\star\|$ is a norm of $\star$ in its own space, and the norm is typically the L^2 norm. The amplitude $A^{\tilde{n}}$ is a measure of how important is the $\tilde{n}$th mode, hence, it is used as a stopping criterion for the enrichment of the solution. Moreover, the alternated direction fixed point loop could be controlled, at each iteration "`iter`", by comparing a given tolerance "`tol`" with a stationary measure of the amplitude

$$\frac{|A^{\tilde{n}}_{\mathtt{iter}} - A^{\tilde{n}}_{\mathtt{iter}-1}|}{|A^{\tilde{n}}_{\mathtt{iter}-1}|} < \mathtt{tol}. \tag{3.8}$$

The details of and deeper insight on PGD algorithms could be found in [9, 49, 51, 77] and the references therein. In order to be able to solve the problem computationally, we need to recast our problem in a tensorial format. Once the functional space is discretized, we could reformulate our problem in a tensorial format that is equivalent to the functional format [51]. The standard discretization of the parametric PDE, typically with FEM, results in a system of linear equations to be solved that has the following algebraic form

$$\boldsymbol{K}(\boldsymbol{\mu})\boldsymbol{d}(\boldsymbol{\mu}) = \boldsymbol{f}(\boldsymbol{\mu}), \tag{3.9}$$

where $\boldsymbol{\mu}$ is a generic vector collecting different kinds of parameters that are considered as extra coordinates in the problem, $\boldsymbol{K}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathtt{n_d} \times \mathtt{n_d}}$ is the generalized stiffness matrix,

$\boldsymbol{d}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathtt{n_d}}$ is the vector of generalized displacements that we are solving for, and $\boldsymbol{f}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathtt{n_d}}$ is the generalized vector of forces. Thus, all these fields are taking values in a multiparametric space $I_{\boldsymbol{\theta}}$. This equation is of algebraic nature and is already discretized in space. Note that from now on, for the sake of notation clarity, the generic vector of parameters $\boldsymbol{\mu}$ is replaced with the vector $\boldsymbol{\theta}$ because all our parameters have the same nature, i.e. are fibre orientations. The solution of Eq. (3.9) is based on the weighted residuals approach and is introduced as follows,

$$\boldsymbol{R}\big(\boldsymbol{d}(\boldsymbol{\theta})\big) := \boldsymbol{f}(\boldsymbol{\theta}) - \boldsymbol{K}(\boldsymbol{\theta})\boldsymbol{d}(\boldsymbol{\theta}). \tag{3.10}$$

Therefore we could state that $\boldsymbol{d}(\boldsymbol{\theta})$ is a solution for Eq. (3.9) if and only if

$$\int_{I_1} \cdots \int_{I_{\mathtt{n_p}}} \boldsymbol{U}^*(\boldsymbol{\theta})\boldsymbol{R}\big(\boldsymbol{d}(\boldsymbol{\theta})\big)\, d\theta_{\mathtt{n_p}} \dots d\theta_1 = 0, \tag{3.11}$$

where $\boldsymbol{U}^*$ is a test function and the integration is performed for $\mathtt{n_p}$ parameters. The separated form of Eq. (3.11) is solved with the PGD solver. To perform the integration, we need to define first a range or interval for the parameters such that $\theta_i \in I_i \subset \mathbb{R}$. Once this is chosen, we explicitly discretize those intervals by choosing a number of points inside the interval where the solution will be sought. After we obtain our generalized solution, if we desire to particularize our solution to a parametric value that is between two discretization points, a linear interpolation is performed. The encapsulated PGD concept that takes as input a separated stiffness matrix $\boldsymbol{K}(\boldsymbol{\theta})$ and $\boldsymbol{f}(\boldsymbol{\theta})$ and gives as output the vector of generalized displacements $\boldsymbol{d}(\boldsymbol{\theta})$ is introduced next.

3.4 Encapsulated PGD

The PGD approach is introduced here following the *encapsulated* concept presented in [50, 51], where it provides tools that directly produce the computational vademecums for the high-dimensional tensor data. The encapsulated PGD concept allows to define PGD objects, which are quantities defined in a multidimensional setting representing multiparametric functions, and it provides a toolbox[1] of algebraic routines to directly operate with these objects. Thus, the general methodology permits the performance of non-trivial operations (e.g. solving linear systems of equations, compression, etc...) for multidimensional tensors, shown in [51]. For example, the input parametric matrix

[1] Publicly available at https://git.lacan.upc.edu/zlotnik/algebraicPGDtools

$\boldsymbol{K}(\boldsymbol{\theta})$ in (2.25) has to be provided in (or approximated by) a separated form, that is

$$\boldsymbol{K}(\boldsymbol{\theta}) \approx \boldsymbol{K}^{\mathtt{sep}}(\boldsymbol{\theta}) = \sum_{k=1}^{\mathtt{n_k}} \boldsymbol{K}^k \prod_{j=1}^{\mathtt{n_p}} \varphi_j^k(\theta_j), \tag{3.12}$$

where superscript "sep" indicates that the quantity (in this case the matrix) is stored in a separated form. For each term $k = 1, \dots, \mathtt{n_k}$, matrices $\boldsymbol{K}^k$ and functions φ_j^k, for $j = 1, \dots, \mathtt{n_p}$ are the spatial and the parametric modes, respectively, describing the parametric dependence of the global stiffness matrix using an affine decomposition with $\mathtt{n_k}$ terms.

One of the routines in the toolbox is the linear solver, having as input $\boldsymbol{K}^{\mathtt{sep}}(\boldsymbol{\theta})$ and $\boldsymbol{f}$ (possibly $\boldsymbol{f}^{\mathtt{sep}}(\boldsymbol{\theta})$) and yielding as output a separable approximation to the unknown vector of generalized displacements $\boldsymbol{d}(\boldsymbol{\theta})$, namely,

$$\begin{aligned} \boldsymbol{d}(\boldsymbol{\theta}) \approx \boldsymbol{d}^n_{\mathtt{PGD}}(\boldsymbol{\theta}) &= \sum_{m=1}^{n} \beta^m \boldsymbol{d}^m \prod_{j=1}^{\mathtt{n_p}} G_j^m(\theta_j) \\ &= \boldsymbol{d}^{n-1}_{\mathtt{PGD}}(\boldsymbol{\theta}) + \beta^n \boldsymbol{d}^n G_1^n(\theta_1) G_2^n(\theta_2) ... G^n_{\mathtt{n_p}}(\theta_{\mathtt{n_p}}), \end{aligned} \tag{3.13}$$

where $\boldsymbol{d}^n_{\mathtt{PGD}}$ is a separated approximation with n terms; $\boldsymbol{d}^m$ is the spatial mode, and G_j^m are the parametric modes where $m = 1, \dots, n$ and $j = 1, \dots, \mathtt{n_p}$. Modes $\boldsymbol{d}^m$ and G_j^m are normalized and β^m collects the amplitude of each term. Amplitude β^m accounts for the importance of term m and is also used to decide when to stop the greedy algorithm (one stops computing new terms once β^m is small enough, with respect to β^1).

Often, the PGD solution has redundant information as orthogonality between successive terms is not enforced; whereas it is enforced, for instance, in the Singular Value Decomposition (SVD can be seen as a particular case of PGD). The PGD compression is a methodology that post-processes any PGD object, aiming to alleviate the excess of PGD terms associated with redundant information (reduce a too large value of n in (3.13)). It consists in least-squares approximation following the same PGD philosophy, see [50, 56, 63]. In a nutshell, for any solution provided by the PGD solver like $\boldsymbol{d}^n_{\mathtt{PGD}}$ (as the solution of (3.13)) the goal is to find a PGD-type approximation $\boldsymbol{d}^{\mathtt{n_c}}_{\mathtt{com}}$ such that the following discrepancy is minimized,

$$\|\boldsymbol{d}^{\mathtt{n_c}}_{\mathtt{com}} - \boldsymbol{d}^n_{\mathtt{PGD}}\|_{L^2(I_{\boldsymbol{\theta}})} = \int_{I_1} \cdots \int_{I_{\mathtt{n_p}}} (\boldsymbol{d}^{\mathtt{n_c}}_{\mathtt{com}} - \boldsymbol{d}^n_{\mathtt{PGD}})^2 \, d\theta_{\mathtt{n_p}} \dots d\theta_1. \tag{3.14}$$

Note that the number of terms, $\mathtt{n_c}$, in the compressed solution $\boldsymbol{d}^{\mathtt{n_c}}_{\mathtt{com}}$ is expected to be significantly lower than the original one ($\mathtt{n_c} \ll n$).

3.5 Separation of input for PGD solver

As indicated in the previous section, the input of the encapsulated PGD routines is made of separated PGD objects, as the stiffness matrix described in (3.12). In the present case, the parametric dependence of the input matrix $\boldsymbol{K}(\boldsymbol{\theta})$ on the parameters θ_i, $i = 1, \ldots, \mathtt{n_p}$ arises from the parametric dependence of the elasticity tensor $\boldsymbol{\mathcal{C}}(\boldsymbol{\theta})$; which depends on the value of the fiber orientation at the material point where it is evaluated. A separated representation of $\boldsymbol{\mathcal{C}}(\boldsymbol{\theta})$ is required in order to build up a separated representation of matrix $\boldsymbol{K}(\boldsymbol{\theta})$ as in (3.12).

Recalling Section 2.3, it is assumed that the subdomains Ω_i, $i = 1, \ldots, \mathtt{n_p}$, where the angle of the fiber orientation is θ_i, do cover the whole domain Ω; that is $\bar{\Omega} = \bigcup_{i=1}^{\mathtt{n_p}} \bar{\Omega}_i$. Thus, the elasticity tensor depends at each point $\boldsymbol{x} \in \Omega_i$ on the parameter θ_i, and it is expressed in a separable format as:

$$\boldsymbol{\mathcal{C}}(\theta_i) = \sum_{\ell=1}^{\mathtt{n_t}} \boldsymbol{\mathcal{C}}^{\ell} \prod_{j=1}^{\mathtt{n_p}} \phi_j^{\ell,i}(\theta_j), \tag{3.15}$$

where the fact that $\boldsymbol{\mathcal{C}}(\theta_i)$ depends only on θ_i results in the condition $\phi_j^{\ell,i}(\theta_j) \equiv 1$ for $j \neq i$, see Appendix B for details.

Moreover, any point $\boldsymbol{x}$ belonging to some element $\Omega^e \subset \Omega_i$, such that the element index, e, $e = 1, \ldots, \mathtt{n_{el}}$, is in relation with subdomain index i.

This formal convention identifying element e with material subdomain i allows replacing (3.15) in the expression for $\boldsymbol{K}^e(\theta_i)$ provided by (2.24), and this results in

$$\boldsymbol{K}^e(\theta_i) = \sum_{\ell=1}^{\mathtt{n_t}} \Big[\int_{\Omega_e} \boldsymbol{B}^{e\mathsf{T}} \boldsymbol{\mathcal{C}}^{\ell} \boldsymbol{B}^e d\Omega \Big] \prod_{j=1}^{\mathtt{n_p}} \phi_j^{\ell,i}(\theta_j). \tag{3.16}$$

Then, assembling the local matrices as indicated in (2.6), one gets

$$\begin{aligned} \boldsymbol{K}(\theta_1, \theta_2, ..., \theta_{\mathtt{n_p}}) &= \sum_{e=1}^{\mathtt{n_{el}}} \boldsymbol{L}^{e\mathsf{T}} \boldsymbol{K}^e(\theta_i) \boldsymbol{L}^e \\ &= \sum_{e=1}^{\mathtt{n_{el}}} \sum_{\ell=1}^{\mathtt{n_t}} \Big[\int_{\Omega_e} \boldsymbol{L}^{e\mathsf{T}} \boldsymbol{B}^{e\mathsf{T}} \boldsymbol{\mathcal{C}}^{\ell} \boldsymbol{B}^e \boldsymbol{L}^e \, d\Omega \Big] \prod_{j=1}^{\mathtt{n_p}} \phi_j^{\ell,i}(\theta_j), \end{aligned} \tag{3.17}$$

which provides a separable expression for $\boldsymbol{K}(\boldsymbol{\theta})$ that is used as input for the encapsulated PGD routines. In particular, the linear solver for algebraic equations provides as output $\boldsymbol{d}^n_{\mathtt{PGD}}(\boldsymbol{\theta})$.

In the following is presented the pseudo-algorithm for the computation of the separated stiffness matrix $\boldsymbol{K}$:

Algorithm 3.1: Pseudo-code of the FE loop to build the separated stiffness matrix for the PGD routines

Input : Mesh coordinates, FE connectivities, number of elements, parametric subdomains Ω_i

Output: Separated stiffness matrix $\boldsymbol{K}$

(1) Evaluate the separated terms of the elasticity tensor

for $e = 1, ..., \mathtt{n}_{el}$ **do**

(1.1) Extract element coordinates and connectivities

if $e \in \Omega_i$ **then**

- Assign θ_i to element e with $i = 1, ...\mathtt{n_p}$

end

for $\ell = 1, ..., \mathtt{n}_t$ **do**

- Compute $(\boldsymbol{K}^e)^\ell$ like in Eq. (3.16)
- Assign local $(\boldsymbol{K}^e)^\ell$ to global $\boldsymbol{K}^k(\theta_1, \theta_2, ..., \theta_{\mathtt{n_p}})$ and $k = (\ell+(i-1)\cdot\mathtt{n_t}), ..., (i\cdot\mathtt{n_t})$

3.6 Post-process and sensitivities

Once the parametric solution $\boldsymbol{d}^n_{\mathtt{PGD}}(\boldsymbol{\theta})$ is obtained in the form of a computational vademecum (3.13), it has to be used to compute the parametric expressions of the failure index $\mathcal{I}_{\mathtt{f}}$, see (2.30), and the safety factor, λ_s, see (2.42). In order to solve the optimization problems (2.45) and (2.46) with gradient-based methods, the sensitivities (gradients and Hessian matrices with respect to the parameters) need to be computed.

In a first step, the strain tensor has to be computed as a postprocess of the parametric displacements $\boldsymbol{d}^n_{\mathtt{PGD}}(\boldsymbol{\theta})$. In practice, the strain field is computed in a set of $\mathtt{n_g}$ points in

domain Ω, typically the integration points of the finite element mesh which are indexed with $g = 1, \ldots, \mathtt{n_g}$. At each of these points, the strain tensor is a vector of 6 components (using Voigt's notation), which is a linear function of the displacement field, see (2.2). Thus, globally the strain field is described by a $6 \times \mathtt{n_g}$ matrix depending on the parameters, $\boldsymbol{\varepsilon}(\boldsymbol{\theta})$. Each column of this matrix is a 6×1 vector denoted $\boldsymbol{\varepsilon}_g(\boldsymbol{\theta})$ and represents the strain tensor at point g.

Assuming that point g is in element Ω^e, the strain at point g is a linear output of the overall displacements $\boldsymbol{d}$, namely,

$$\boldsymbol{\varepsilon}_g = \boldsymbol{B}_g^e \boldsymbol{L}^e \boldsymbol{d},$$

where $\boldsymbol{B}_g^e$ is matrix $\boldsymbol{B}^e$ (same as in equation (2.12)) evaluated at point g and $\boldsymbol{L}^e$ is the Boolean operator localizing global displacements to element degrees of freedom. Consequently, using the parametric expression of the displacements in (3.13) results in the following expression for the parametric strains at point g:

$$\boldsymbol{\varepsilon}_g(\boldsymbol{\theta}) = \sum_{m=1}^{n} \beta^m \boldsymbol{\varepsilon}_g^m \prod_{j=1}^{\mathtt{n_p}} G_j^m(\theta_j), \tag{3.18}$$

where $\boldsymbol{\varepsilon}_g^m = \boldsymbol{B}_g^e \boldsymbol{L}^e \boldsymbol{d}^m$.

The format of the strain field $\boldsymbol{\varepsilon}(\boldsymbol{\theta})$, that is a $6 \times \mathtt{n_g}$ matrix, with columns $\boldsymbol{\varepsilon}_g(\boldsymbol{\theta})$ representing strains at point g, is replicated to describe the stresses. Thus, stresses are stored in a $6 \times \mathtt{n_g}$, $\boldsymbol{\sigma}(\boldsymbol{\theta})$, such that each column of this matrix is a 6×1 vector $\boldsymbol{\sigma}_g(\boldsymbol{\theta})$ representing the stress tensor at point g.

The relation between strains and stresses at point g is given by the corresponding elasticity tensor $\boldsymbol{\mathcal{C}}$, see (2.2). Thus, the stresses at point g, $\boldsymbol{\sigma}_g(\boldsymbol{\theta}) = \boldsymbol{\mathcal{C}}(\theta_i)\boldsymbol{\varepsilon}_g(\boldsymbol{\theta})$ become, using (3.15) and (3.18),

$$\boldsymbol{\sigma}_g(\boldsymbol{\theta}) = \sum_{m=1}^{n} \sum_{\ell=1}^{\mathtt{n_t}} \beta^m \boldsymbol{\mathcal{C}}^\ell \boldsymbol{\varepsilon}_g^m \prod_{j=1}^{\mathtt{n_p}} \phi_j^{\ell,i}(\theta_j) G_j^m(\theta_j), \tag{3.19}$$

where it is worth noting that, similarly as in the previous equations, index i is associated with index g, in the sense that it is assumed that point g is in subdomain Ω_i. Sorting the terms with a single index $q = 1, \ldots, n \cdot \mathtt{n_t}$ instead of the two indices m and ℓ, (3.19) is rewritten as:

$$\boldsymbol{\sigma}_g(\boldsymbol{\theta}) = \sum_{q=1}^{n \cdot \mathtt{n_t}} \bar{\beta}^q \boldsymbol{\sigma}_g^q \prod_{j=1}^{\mathtt{n_p}} Q_j^q(\theta_j). \tag{3.20}$$

It is assumed that there is an explicit association between a pair (m, ℓ) and index q (for instance $q = m + (\ell - 1) \cdot n$); $\boldsymbol{\sigma}_g^q$ is equal to $\boldsymbol{\mathcal{C}}^\ell \boldsymbol{\varepsilon}_g^m$ divided by its norm, $Q_j^q(\theta_j)$ is the product $\phi_j^{\ell,i}(\theta_j) G_j^m(\theta_j)$ also normalized; and $\bar{\beta}^q$ collects the product of β^m and the normalization factors.

In the remainder of the present section, a similar strategy is employed to compute the parametric dependence of $\mathcal{I}_{\mathtt{f}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big)$, see (2.30) in Section 2.4.2. A parametric separated expression of the transformation matrix $\boldsymbol{T}(\theta_i)$ is needed as a first step to compute the parametric expression for the strength tensor and vector, $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{F}$, see (2.43), namely,

$$\boldsymbol{T}(\theta_i) = \sum_{r=1}^{\mathtt{n_r}} \boldsymbol{T}^r \prod_{j=1}^{\mathtt{n_p}} Z_j^{r,i}(\theta_j), \tag{3.21}$$

where $\mathtt{n_r}$ is the number of terms required to express the transformation matrix in a separated fashion, and similar to the definition of $\phi_j^{\ell,i}$ in (3.15), $Z_j^{r,i}(\theta_j) \equiv 1$ for $i \neq j$. The explicit expressions of all the terms are given in Appendix B.

Using (3.21) in (2.43), results in

$$\boldsymbol{\mathcal{F}}(\theta_i) = \sum_{s=1}^{\mathtt{n_r}} \sum_{r=1}^{\mathtt{n_r}} (\boldsymbol{T}^{r\mathsf{T}} \boldsymbol{\mathcal{F}}_0 \boldsymbol{T}^s) \prod_{j=1}^{\mathtt{n_p}} Z_j^{r,i}(\theta_j) Z_j^{s,i}(\theta_j). \tag{3.22}$$

Analogously as with $\boldsymbol{\sigma}_g$, (3.22) is rewritten using a single index notation (index pair (r,s) is mapped into a single index p, $p = 1, \ldots, \mathtt{n_r^2}$), that is,

$$\boldsymbol{\mathcal{F}}(\theta_i) = \sum_{p=1}^{\mathtt{n_r^2}} \bar{\alpha}^p \boldsymbol{\mathcal{F}}^p \prod_{j=1}^{\mathtt{n_p}} P_j^{p,i}(\theta_j). \tag{3.23}$$

The same is carried out for $\boldsymbol{F}$ and results in

$$\boldsymbol{F}(\theta_i) = \boldsymbol{T}^{\mathsf{T}}(\theta_i) \boldsymbol{F}_0 = \sum_{r=1}^{\mathtt{n_r}} \tilde{\alpha}^r (\boldsymbol{T}^{r\mathsf{T}} \boldsymbol{F}_0) \prod_{j=1}^{\mathtt{n_p}} Z_j^{r,i}(\theta_j). \tag{3.24}$$

The failure index given in (2.30) is divided in two terms, one linear and one quadratic, namely,

$$\mathcal{I}_{\mathtt{L}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) = \boldsymbol{\sigma}_g^{\mathsf{T}} \boldsymbol{F}(\theta_i) \text{ and } \mathcal{I}_{\mathtt{Q}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) = \boldsymbol{\sigma}_g^{\mathsf{T}} \boldsymbol{\mathcal{F}}(\theta_i) \boldsymbol{\sigma}_g. \tag{3.25}$$

The expression for $\boldsymbol{\sigma}_g$ and $\boldsymbol{\mathcal{F}}$ in (3.20) and (3.23) are used in (3.25) to obtain the following expression for $\mathcal{I}_{\mathtt{Q}}$

$$\mathcal{I}_{\mathtt{Q}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) = \sum_{p=1}^{\mathtt{n_r^2}} \sum_{w=1}^{n\cdot\mathtt{n_t}} \sum_{q=1}^{n\cdot\mathtt{n_t}} \bar{\alpha}^p \bar{\beta}^w \bar{\beta}^q (\boldsymbol{\sigma}_g^{q\,\mathsf{T}} \boldsymbol{\mathcal{F}}^p \boldsymbol{\sigma}_g^w) \prod_{j=1}^{\mathtt{n_p}} P_j^{p,i}(\theta_j) Q_j^{w,i}(\theta_j) Q_j^{q,i}(\theta_j). \tag{3.26}$$

Again, transforming the three indices (p, w, q) into one index b, $b = 1, \dots, \mathtt{n_r^2} n^2 \mathtt{n_t^2}$, the following expression is obtained

$$\mathcal{I}_{\mathtt{Q}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) = \sum_{b=1}^{\mathtt{n_r^2} n^2 \mathtt{n_t^2}} \tilde{\gamma}^b \tilde{A}_g^b \prod_{j=1}^{\mathtt{n_p}} \tilde{H}_j^{b,i}(\theta_j), \tag{3.27}$$

where $\tilde{A}_g^b$ and $\tilde{H}_j^{b,i}(\theta_j)$ are the normalized versions of $\boldsymbol{\sigma}_g^{q\,\mathsf{T}} \boldsymbol{\mathcal{F}}^p \boldsymbol{\sigma}_g^w$ and $P_j^{p,i}(\theta_j) Q_j^{w,i}(\theta_j) Q_j^{q,i}(\theta_j)$ respectively, and $\tilde{\gamma}^b$ collects the amplitude $\bar{\alpha}^p \bar{\beta}^w \bar{\beta}^q$ and all the normalization factors.

Analogously, for the linear part of the failure index, $\mathcal{I}_{\mathtt{L}}$, we have

$$\mathcal{I}_{\mathtt{L}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) = \sum_{q=1}^{n \cdot \mathtt{n_t}} \sum_{r=1}^{\mathtt{n_r}} \bar{\beta}^q \tilde{\alpha}^r (\boldsymbol{\sigma}_g^{q\,\mathsf{T}} \boldsymbol{F}^r) \prod_{j=1}^{\mathtt{n_p}} Q_j^{q,i}(\theta_j) Z_j^{r,i}(\theta_j), \tag{3.28}$$

that in a single index format (associating (q, r) to v, $v = 1, \dots, n \cdot \mathtt{n_t} \cdot \mathtt{n_r}$) results in

$$\mathcal{I}_{\mathtt{L}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) = \sum_{v=1}^{n \cdot \mathtt{n_t} \cdot \mathtt{n_r}} \hat{\gamma}^v \hat{A}_g^v \prod_{j=1}^{\mathtt{n_p}} \hat{H}_j^{v,i}(\theta_j), \tag{3.29}$$

where we define $\mathtt{n_L} := \mathtt{n_r} \cdot n \cdot \mathtt{n_t}$ to ease the notation. The expression for the failure index $\mathcal{I}_{\mathtt{f}}$ is readily recovered by summing up (3.27) and (3.29), that is,

$$\mathcal{I}_{\mathtt{f}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) = \mathcal{I}_{\mathtt{Q}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) + \mathcal{I}_{\mathtt{L}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big) = \sum_{f=1}^{\mathtt{n_Q}+\mathtt{n_L}} \gamma^f A_g^f \prod_{j=1}^{\mathtt{n_p}} H_j^{f,i}(\theta_j), \tag{3.30}$$

where the quantities γ^f, A_g^f and $H_j^{f,i}(\theta_j)$ are equal to the ones in (3.27) or (3.29) depending on the index f,

$$\gamma^f,\, A_g^f,\, H_g^f = \begin{cases} \tilde{\gamma}^f, \tilde{A}_g^f, \tilde{H}_g^f & \text{if} f \leq \mathtt{n_Q} \\ \hat{\gamma}^{f-\mathtt{n_Q}}, \hat{A}_g^{f-\mathtt{n_Q}}, \hat{H}_g^{f-\mathtt{n_Q}} & \text{if } f > \mathtt{n_Q} \end{cases},$$

and, for the sake of shortening the writing, the number of PGD terms needed to express $\mathcal{I}_{\mathtt{Q}}$ is introduced as $\mathtt{n_Q} := \mathtt{n_r^2} n^2 \mathtt{n_t^2}$.

Once $\mathcal{I}_{\mathtt{f}}\big(\boldsymbol{\sigma}_g(\boldsymbol{\theta})\big)$ is obtained in the form of (3.30), the multiple queries required to solve the optimization problem defined in (2.45) (or in (2.46)) may be performed very fast, as a simple post-processing.

Moreover, an additional advantage of the PGD solutions is that it provides a solution depending on the parameters explicitly allowing the computation of the derivatives of the objective function provided by (3.30) in a straight-forward way. That is, the sensitivities needed in the implementation of the gradient-based optimization methods.

At any sampling point g, the gradient of the failure index is denoted as $\nabla_\theta \mathcal{I}_{\mathtt{f}}(\boldsymbol{\theta})$ and contains all partial derivatives of $\mathcal{I}_{\mathtt{f}}$ with respect to θ_k, for $k = 1, \ldots, \mathtt{n_p}$. Using the expression (3.30), these derivatives read

$$\frac{\partial \mathcal{I}_{\mathtt{f}}(\boldsymbol{\theta})}{\partial \theta_k} = \sum_{f=1}^{\mathtt{n_Q}+\mathtt{n_L}} \gamma^f A_g^f \left[\frac{dH_k^{f,i}}{d\theta_k}(\theta_k) \right] \prod_{j \neq k}^{\mathtt{n_p}} H_j^{f,i}(\theta_j). \tag{3.31}$$

Moreover, for the strategies requiring the Hessian matrix, all its components consist in second order derivatives, which are readily computed in a similar fashion. In particular, for $k \neq \tilde{k}$, one has

$$\frac{\partial^2 \mathcal{I}_{\mathtt{f}}(\boldsymbol{\theta})}{\partial \theta_k \partial \theta_{\tilde{k}}} = \sum_{f=1}^{\mathtt{n_Q}+\mathtt{n_L}} \gamma^f A_g^f \left[\frac{dH_k^{f,i}}{d\theta_k}(\theta_k) \, \frac{dH_{\tilde{k}}^{f,i}}{d\theta_{\tilde{k}}}(\theta_{\tilde{k}}) \right] \prod_{j \neq k, \tilde{k}}^{\mathtt{n_p}} H_j^{f,i}(\theta_j) \tag{3.32}$$

and

$$\frac{\partial^2 \mathcal{I}_{\mathtt{f}}(\boldsymbol{\theta})}{\partial {\theta_k}^2} = \sum_{f=1}^{\mathtt{n_Q}+\mathtt{n_L}} \gamma^f A_g^f \left[\frac{d^2 H_k^{f,i}}{d{\theta_k}^2}(\theta_k) \right] \prod_{j \neq k}^{\mathtt{n_p}} H_j^{f,i}(\theta_j). \tag{3.33}$$

The derivatives of the parametric modes $H_k^{f,i}$ in equation (3.31) are performed numerically. Typically, the modes are stored in terms of vectors of nodal variables, following the FE philosophy. Thus, assuming that the nodal values of function $H_k^{f,i}(\theta_k)$ are collected in vector $\boldsymbol{h}$, the question is how to compute the derivatives. In other words, assuming that $\frac{dH_k^{f,i}}{d\theta_k}$ is stored in the same fashion in the vector of nodal values $\boldsymbol{g}$, how to compute $\boldsymbol{g}$ from $\boldsymbol{h}$? Note that the parametric range for θ_k, I_k, is typically 1D (a subset of $\mathbb{R}$) and therefore explicit numerical differentiation node-wise is straightforward. A more consistent approach is based on the least-squares projection on the initial discrete functional space of the sectional approximation. Recall that the adjective sectional is used in this context to refer to operations in a single parametric dimension. In summary, the derivation of the function described by $\boldsymbol{h}$ consists in computing $\boldsymbol{g}$ such that

$$\boldsymbol{M}\boldsymbol{g} = \boldsymbol{D}\boldsymbol{h}, \tag{3.34}$$

where $\boldsymbol{M}$ is the sectional mass matrix and $\boldsymbol{D}$ is the sectional gradient matrix. Both $\boldsymbol{M}$ and $\boldsymbol{D}$ are very simple matrices (in the usual case of being I_k a 1D sectional domain discretized with linear finite elements, they are tridiagonal matrices). In a more general case, they result from assembling elemental matrices having the form,

$$\boldsymbol{M}^{\boldsymbol{e}} = \int_{I_k} \tilde{\boldsymbol{N}}^{e\,\mathsf{T}} \tilde{\boldsymbol{N}}^e \, d\theta \text{ and } \boldsymbol{D}^{\boldsymbol{e}} = \int_{I_k} \tilde{\boldsymbol{N}}^e \frac{d\tilde{\boldsymbol{N}}^e}{d\theta} \, d\theta,$$

where $\tilde{\boldsymbol{N}}^e$ is the vector of element shape functions in each element discretizing I_k, that is the parametric counterpart of the shape functions introduced in equation (2.6) for the

space approximation.

Similarly, at any sampling point g, the gradient of the safety factor is denoted as $\nabla_\theta \lambda_s(\boldsymbol{\theta})$ and contains all partial derivatives of $\mathcal{I}_{\mathtt{Q}}$ and $\mathcal{I}_{\mathtt{L}}$ with respect to θ_k, for $k = 1, \ldots, \mathtt{n_p}$. Using the expressions in (3.27) and (3.29), these derivatives read

$$\frac{\partial \mathcal{I}_{\mathtt{Q}}(\boldsymbol{\theta})}{\partial \theta_k} = \sum_{b=1}^{\mathtt{n_Q}} \tilde{\gamma}^b \tilde{A}_g^f \left[\frac{d\tilde{H}_k^{b,i}}{d\theta_k}(\theta_k) \right] \prod_{j \neq k}^{\mathtt{n_p}} \tilde{H}_j^{b,i}(\theta_j) \tag{3.35a}$$

$$\frac{\partial \mathcal{I}_{\mathtt{L}}(\boldsymbol{\theta})}{\partial \theta_k} = \sum_{v=1}^{\mathtt{n_L}} \hat{\gamma}^v \hat{A}_g^f \left[\frac{d\hat{H}_k^{v,i}}{d\theta_k}(\theta_k) \right] \prod_{j \neq k}^{\mathtt{n_p}} \hat{H}_j^{v,i}(\theta_j). \tag{3.35b}$$

Following the quotient rule to evaluate the derivative of the division in (2.42), we obtain the following expression:

$$\frac{\partial \lambda_s(\boldsymbol{\theta})}{\partial \theta_k} = \frac{\mathcal{I}_{\mathtt{Q}} \left[-\dfrac{\partial \mathcal{I}_{\mathtt{L}}(\boldsymbol{\theta})}{\partial \theta_k} + 0.5(\mathcal{I}_{\mathtt{L}}^2 + 4\mathcal{I}_{\mathtt{Q}})^{-1/2} \cdot \left(2\mathcal{I}_{\mathtt{L}} \dfrac{\partial \mathcal{I}_{\mathtt{L}}(\boldsymbol{\theta})}{\partial \theta_k} + 4 \dfrac{\partial \mathcal{I}_{\mathtt{Q}}(\boldsymbol{\theta})}{\partial \theta_k} \right) \right] - \dfrac{\partial \mathcal{I}_{\mathtt{Q}}(\boldsymbol{\theta})}{\partial \theta_k} \left[-\mathcal{I}_{\mathtt{L}} + \sqrt{\mathcal{I}_{\mathtt{L}}^2 + 4\mathcal{I}_{\mathtt{Q}}} \right]}{2\mathcal{I}_{\mathtt{Q}}^2}. \tag{3.36}$$

The sensitivities provided by equations (3.31), (3.33), and (3.36) are extremely useful for gradient-based methods. However, the optimization problem defined in Section 2.5 is in general non-convex and, additionally, gradient-based methods are extremely sensitive to initial guesses leading to a non-convergent iterative procedure. In order to carry out a first global inspection of the parametric domain providing a proper initial guess, it is interesting to consider some evolutionary strategies as Genetic Algorithm (GA), or Simulated Annealing (SA). In a second phase, and starting from a fair initial guess, a gradient-based algorithm is a robust complementary approach converging fast to an accurate solution.

Chapter 4

NUMERICAL EXAMPLES

4.1 Preliminaries

From the previous chapters we already know that PGD is able to obtain all possible solutions for a given parametric PDE and store it in a computational vademecum. The availability of the whole space of solutions, i.e. spatial and parametric spaces, is an attractive feature when solving an optimization problem. Recalling from Chapter 2 in Section 2.5, the objective function of the optimization problem that we would like to solve is highly non-convex, leading to a difficulty in applying a gradient based method since it could easily get stuck in local minima and it usually fails to find the global optimum. Consequently, one would need to solve the problem a very large number of times to converge to a global solution using a heuristic method, such as the genetic algorithms (GA), which is computationally expensive. Generally, when using a standard FE approach, there is no need to compute the solution for the whole parametric space to solve an optimization problem having a small number of design variables (i.e. parameters). However, with the increase of the number of parameters and with the non-convex nature of the objective function (shown in Section 2.5), having the possibility to access the whole parametric space very fast is compelling. With PGD, we are able to browse the solution at particular values of the parameters very fast which allows us to perform plenty of evaluations of the objective function instantaneously.

In the current chapter, we present the solution for the 3D elasticity problem using PGD as described in Chapter 3. We present two main examples to demonstrate the capabilities

and potential of the methodology described in the previous chapters. Both examples are based on composite laminates parameterized with the orientation of the fibers in different subdomains. In the coming examples, we quantify the accuracy of PGD with respect to standard FE, and we discuss different PGD features such as its amplitude resolution, computational cost, and optimization solutions.

The first example in Section 4.2 is a square plate under tensile load having only two parameters where each parameter characterizes a single layer in the domain. Given the simplicity of the problem, it is affordable to compute its solution using standard FE for each point in the parametric space. We will assess the goodness of the PGD by comparing it with the standard FE solution using a global error measure between displacements. Whereas the browsing of the solution in the PGD vademecum at a particular value of the parametric set is extremely fast, the standard FE solution requires a full solve for each parametric value. Note that the computing time to obtain the PGD solution with only two parameters is ~ 2.5 hours; whereas if we would want to compute the solution at every parametric point using standard FE, we would require ~ 6.5 days using the same computer power as in the PGD case. We will also show different loading cases for the same geometry to highlight the optimization results.

The second example (Section 4.3) is a more complicated one involving a plate with a circular hole in the middle. The example is divided into three sub-examples considering two, four, and eight parameters. The domain is subdivided into patches of elements belonging to the $\{O, x, y\}$ plane rather than layers. As the number of parameters increases, the computational burden increases when using standard FE. For instance, in the case of the example with four parameters, the PGD provides a solution in ~ 30 hours and the cost of computing the standard FE solution at every parametric point would be approximately $\sim 10^6$ hours. We will show the optimization results computed using the GA, and their variation with the number of parameters. The chapter is organized as shown in Fig. 4.1.

In both examples, the parameters take values in the range $\theta_i \in I_i = [-90°, 90°]$, for $i = 1, \ldots, \mathtt{n_p}$. Each parametric interval is uniformly discretized by 1° and thus yielding 181 parametric nodes. The material under consideration is the carbon fibre reinforced ABS material, and its characteristics are shown in Appendix C. The tolerance acting as the stopping condition for the alternated directions scheme is set, for both examples, to $\mathtt{tol} = 10^{-6}$. Moreover, the greedy algorithm tolerance which acts as the stopping condition for the modes enrichment loop takes the value $\xi = 10^{-4}$ unless stated otherwise; and it is described through the following amplitudes ratio for the n-th mode,

$$\frac{\beta^n}{\beta^1} < \xi. \tag{4.1}$$

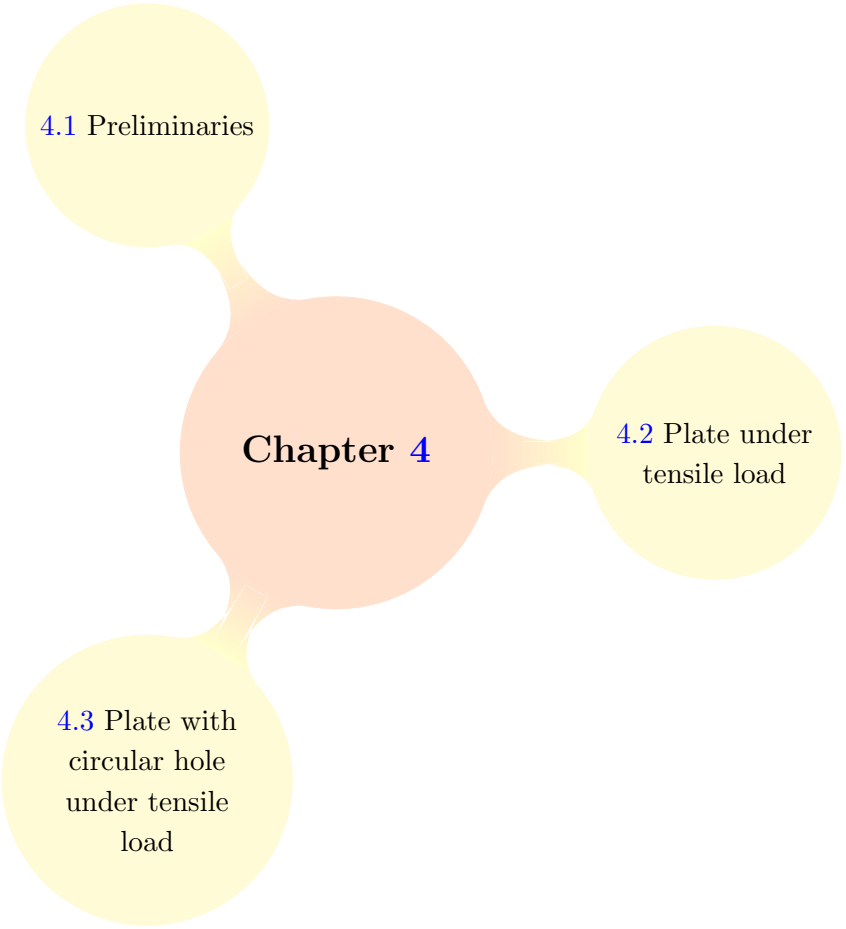


Figure 4.1: Structure of Chapter 4.

It is well known that performing a numerical simulation consists of at least three major steps, namely (a) pre-processing; (b) solving; and (c) post-processing. The workflow followed in this thesis for numerical simulations to find the optimal fibre orientation in composite laminates is briefly summarised in Fig. 4.2.

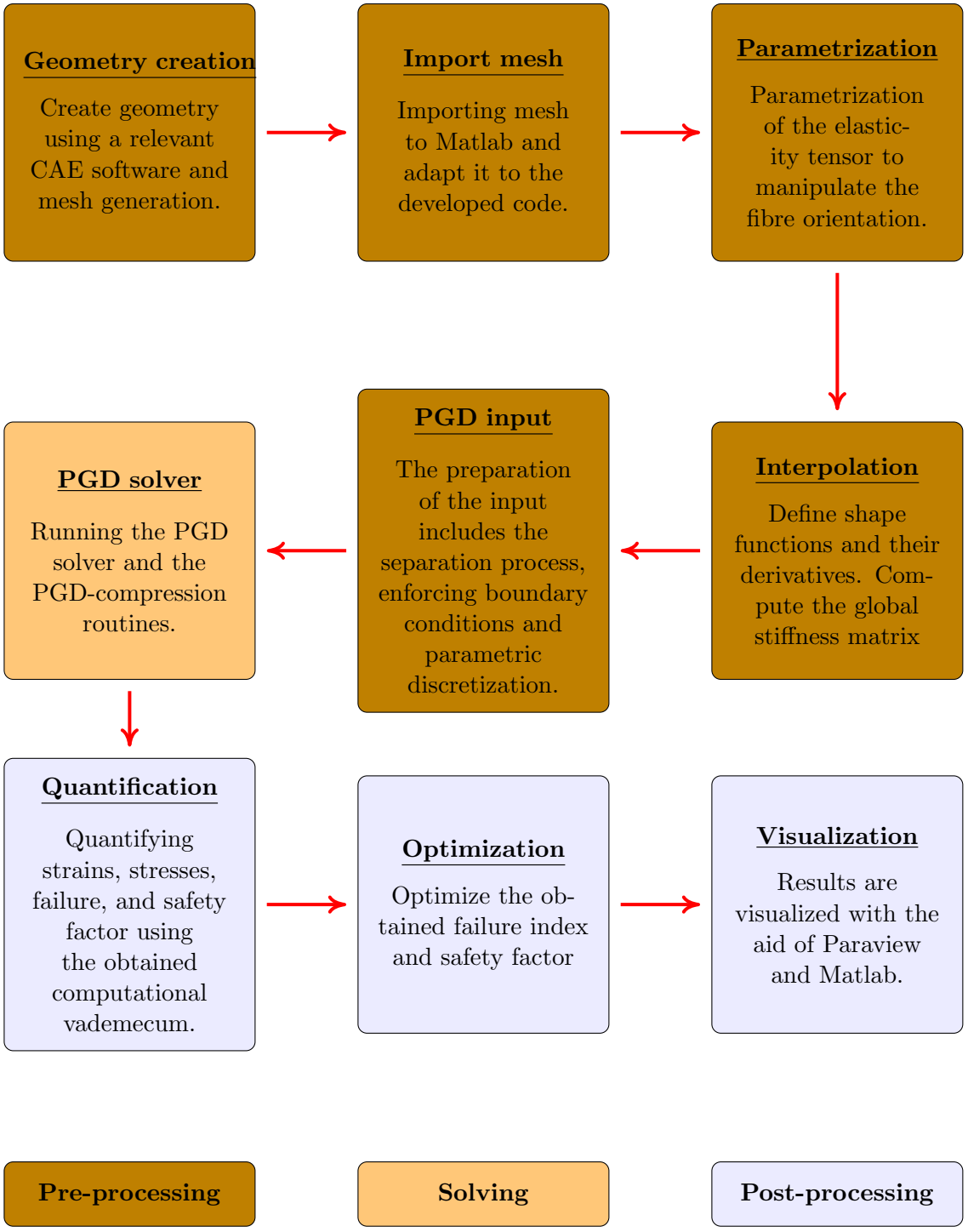


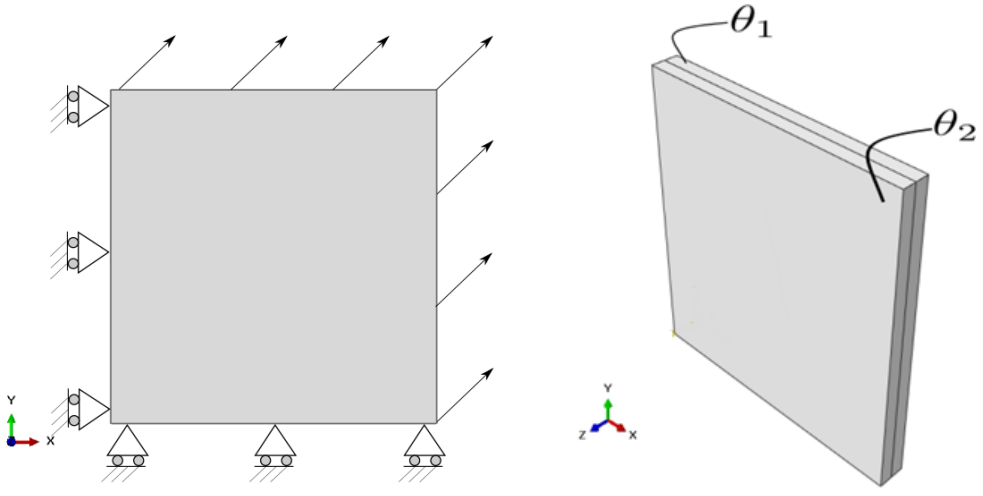
Figure 4.2: Simulation workflow

4.2 Plate under tensile load

This example considers a two-layered composite laminate parameterized by the fibre orientation at each layer. The domain corresponds to a square plate ($60 \times 60 \times 6\ mm^3$) subjected to a tensile load as shown in Fig. 4.3a. Parameters are independent and therefore the material properties at each layer has the form (2.23) and the final separated expression of the operator is that in (3.15).

Parameters θ_1 and θ_2 determine the fibre orientation of layers 1 and 2 respectively as shown in Fig. 4.3b. Parameters take values in the range $\theta_1 \in I_1 = [-90°, 90°]$ and $\theta_2 \in I_2 = [-90°, 90°]$.

The discretization of space involves 800 hexahedral Serendipity elements (4725 nodes) and discretization of both parameters is done with a uniform 1° spaced grid and thus yielding 181 parametric nodes. Note that, despite the parametric space is two dimensional, because of the separated structure of the problem, each parameter dimension is discretized independently as a one dimensional grid. The mechanical properties of the materials are those of carbon fibre reinforced ABS [78] and are described in Appendix C. The plate is under a 45° tensile in-plane load with respect to the x-axis.



(a) Domain top plane (b) Domain 3D view

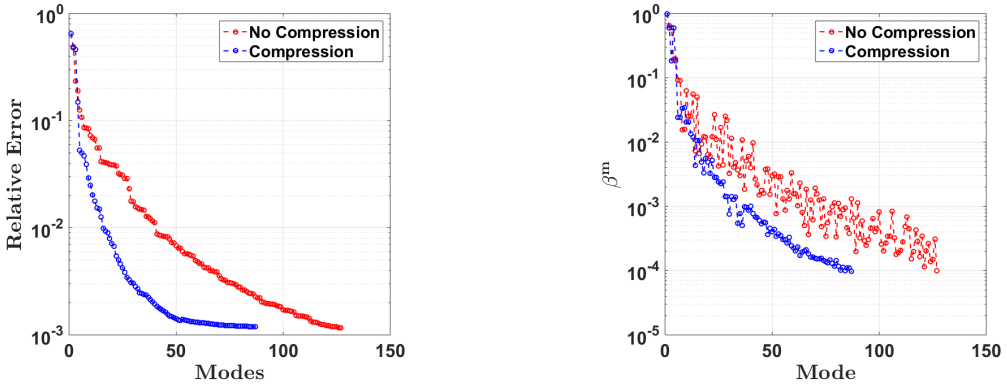
Figure 4.3: Plate under tensile load of 45°

We assess the goodness of the PGD solution by comparing it with the standard FEM solution. The comparison is done by measuring the norm of the difference between the PGD and standard FEM displacements, $\Delta \boldsymbol{d} = \boldsymbol{d}_{\texttt{PGD}} - \boldsymbol{d}_{\texttt{FEM}}$, integrated in space and

parameters as:

$$e = \frac{\sum_{\mathrm{S}_i}\sum_{\mathrm{S}_j} \Delta \boldsymbol{d}^{\mathsf{T}}(\theta_i,\theta_j)\boldsymbol{M}\Delta \boldsymbol{d}(\theta_i,\theta_j)}{\sum_{\mathrm{S}_i}\sum_{\mathrm{S}_j} \boldsymbol{d}^{\mathsf{T}}_{\mathtt{FEM}}(\theta_i,\theta_j)\boldsymbol{M}\boldsymbol{d}_{\mathtt{FEM}}(\theta_i,\theta_j)}, \tag{4.2}$$

where the $\boldsymbol{M}$ matrix is a mass matrix for the space dimension. Note that the error is estimated based on subsets of the parametric grids, $\mathrm{S}_i \in I_1$ and $\mathrm{S}_j \in I_2$. We choose the subsets to reduce the number of FEM problems that is required to be solved for the comparison. Here we use subsets S_i and S_j with one parametric value every 3° instead of every 1° and therefore the solutions are compared at $61 \times 61 = 3721$ instead of $181 \times 181 = 32761$ parametric points. Results of the PGD parametric solution show an excellent agreement with those obtained by standard FEM having errors around 0.1% between both methods with 87 modes in the PGD compressed solution as shown in Fig. 4.4a. It is worth noting that the number of modes in the PGD solution is controlled by the predefined greedy algorithm tolerance $\xi = 10^{-4}$.



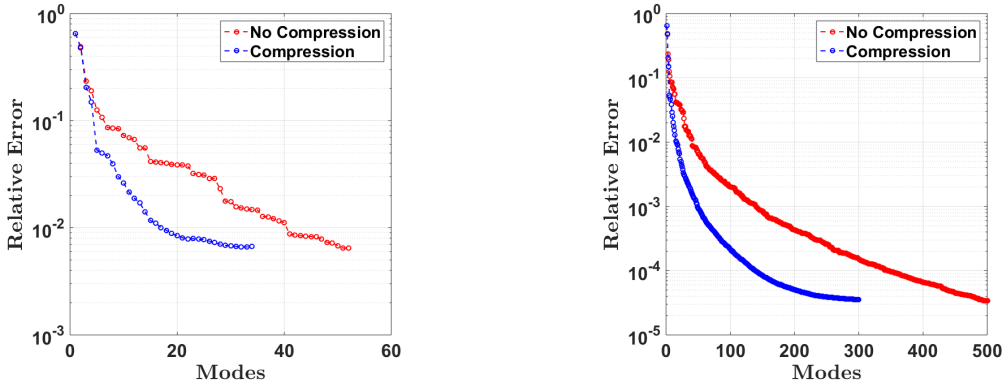
(a) Convergence curve of PGD solution

(b) Amplitude evolution with number of modes

Figure 4.4: Convergence curve and amplitude evolution for plate under tensile load of 45° with greedy tolerance $\xi = 10^{-4}$

In Fig. 4.4a, the convergence curve of the compressed PGD reaches a *plateau* after ~ 70 modes and, after that, the error does not decrease significantly when adding new modes. Moreover, the compressed and uncompressed PGD solutions converge to the same error proving that the compression routine is performing efficiently. The minimum error shown in Fig. 4.4a is controlled by the tolerances imposed in the PGD algorithm ($\xi = 10^{-4}$). Note that, we consider the obtained error ($\sim 0.1\%$ of the standard FE solution) to be an accepted accuracy for engineering purposes. Looser or stricter tolerances produce a plateau in the convergence curve at higher or lower error values respectively (shown in Fig. 4.5). The amplitude of the terms in the PGD solution indicates the importance of each term and it is shown in Fig. 4.4b. There we can see how the importance of each term reduces with the increase of the number of terms. The red curve corresponds to the amplitude evolution of the uncompressed PGD solution, while the blue curve

corresponds to the PGD compressed solution. The PGD-compression is a post-process applied to any PGD object, and it aims to remove redundant information (shown in Section 3.4). This is a standard procedure that projects the solution into the same space using an L^2 projection [51]. In the example shown in Fig. 4.4b, the number of modes is reduced by 31.5%. We could also notice that the PGD compressed solution curve has a smoother behaviour as the redundant information is removed unlike the uncompressed solution that has a lot of oscillations. Note that, the convergence curve Fig. 4.4a reaches stagnation after 70 modes, while the general trend of the amplitude curve in Fig. 4.4b continues decreasing, therefore, implying that the amplitude cannot be used as a direct estimator of the error.

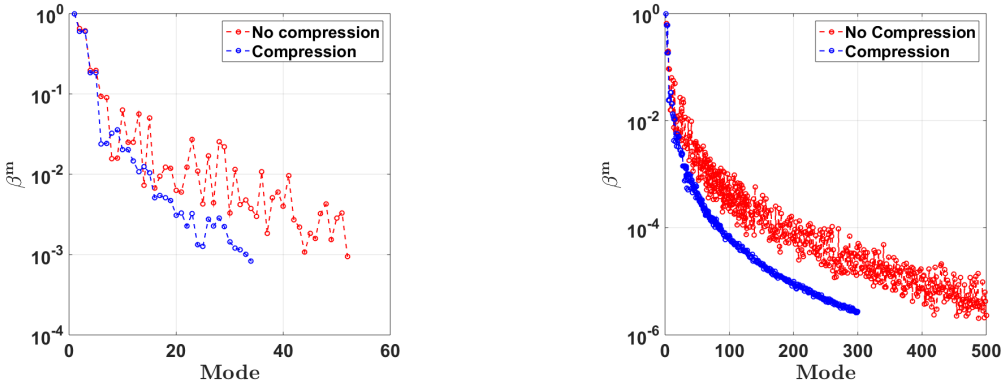


(a) $\xi = 10^{-3}$ (b) $\xi = 10^{-6}$

Figure 4.5: Convergence curves for plate under tensile load of 45° with different greedy algorithm tolerances

In Fig. 4.6, we show the amplitude evolution against the number of modes for different greedy algorithm tolerances. Fig. 4.6a shows the amplitude evolution for a greedy algorithm tolerance of $\xi = 10^{-3}$. The PGD-compression reduces the number of modes by 34.5%. On the other hand, for the strict enrichment tolerance $\xi = 10^{-6}$, we can see the number of terms explode to 500 and 300 for the uncompressed and compressed PGD respectively. It is important to mention that another way to stop the greedy algorithm is to prescribe a maximum number of modes a priori to avoid being stuck if the algorithm is diverging. The convergence curve in Fig. 4.4a shows that the increase of modes does not necessarily add to the accuracy of the problem, i.e., decreasing the relative global error. This suggests that there are factors affecting the global error estimation other than the number of modes which is still an area of investigation [46, 79, 80].

In an attempt to briefly investigate the effect of the spatial mesh size on the amplitude evolution, Fig. 4.7 shows that there is no significant change in the amplitude values nor in the number of modes when a finer or coarser meshes are employed.



(a) $\xi = 10^{-3}$ (b) $\xi = 10^{-6}$

Figure 4.6: Amplitude evolution of uncompressed and compressed PGD solution for plate under tensile 45°for various greedy algorithm tolerances ξ

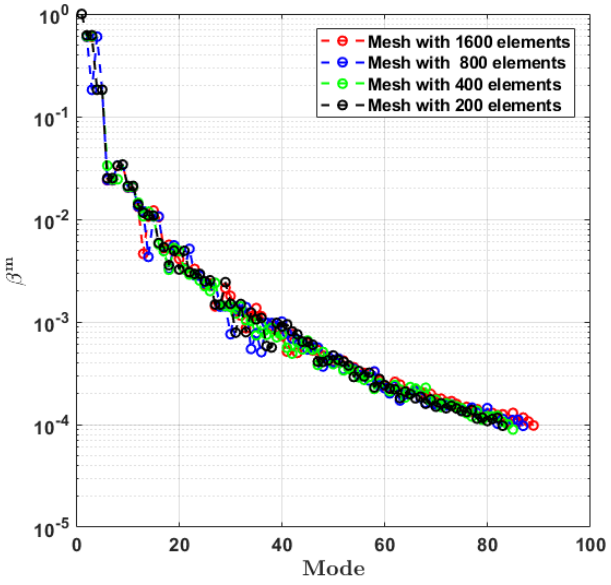
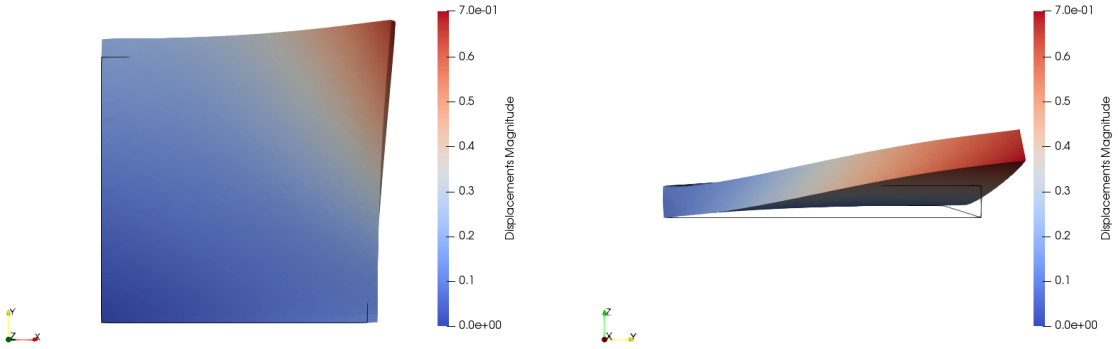


Figure 4.7: Comparison of amplitudes of PGD compressed solution between fine and coarse spatial meshes

The PGD solution consists of a spatial part and a parametric part as demonstrated in Section 3.4. The spatial part is defined by a vector of generalized displacements for each mode, where the parametric part represents the impact of the fibre orientation in each subdomain on the displacements. For the same problem shown previously in Fig. 4.3, we choose a particular value of the set of parameters, namely, $\theta_1 = -45°$ and $\theta_2 = 0°$, which deforms the structure significantly and therefore we could visualize the different spatial modes. Fig. 4.8 shows the magnitude of the reconstructed solution $\boldsymbol{d}_{\texttt{PGD}}$ at $\theta_1 = -45°$ and $\theta_2 = 0°$. Despite the application of the external load only in the xy-plane, we notice

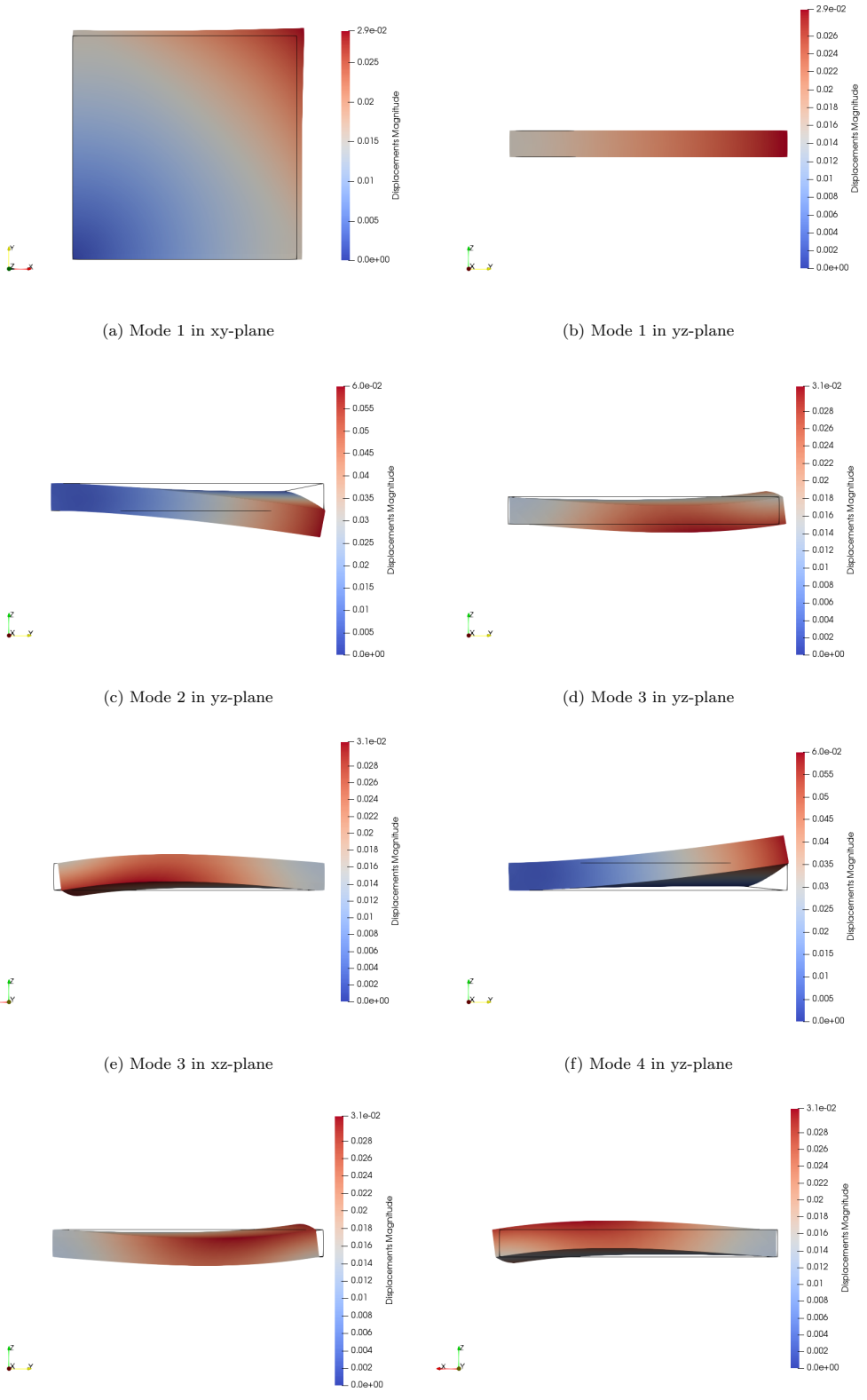
that in addition to the deformation in the xy-plane there is a clear deformation in the xz-plane and the yz-plane that is due to the effect of the significant difference of the fibre orientations in each layer. In the following we show the generalized spatial modes separately to highlight the different modes of deformation.



(a) Displacement magnitude xy-plane

(b) Displacement magnitude yz-plane

Figure 4.8: Displacement magnitude for $\theta_1 = -45°$ and $\theta_2 = 0°$



(a) Mode 1 in xy-plane

(b) Mode 1 in yz-plane

(c) Mode 2 in yz-plane

(d) Mode 3 in yz-plane

(e) Mode 3 in xz-plane

(f) Mode 4 in yz-plane

(g) Mode 5 in yz-plane

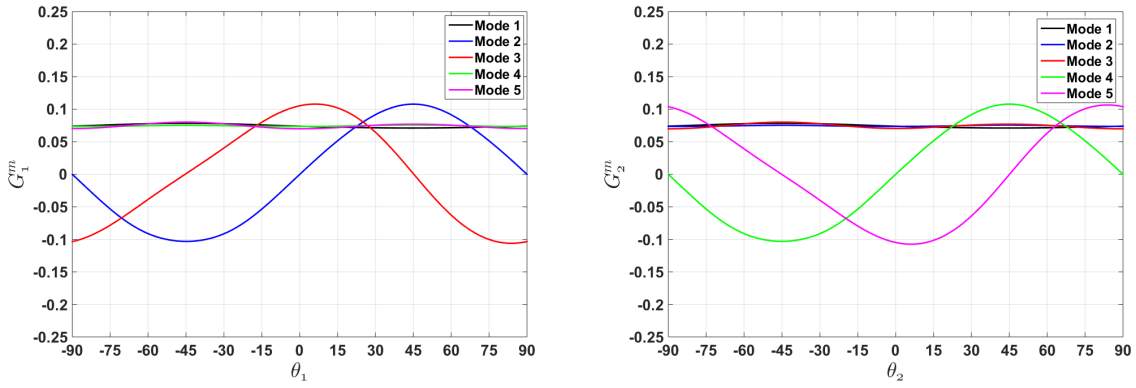
(h) Mode 5 in xz-plane

Figure 4.9: Normalized spatial modes for $\theta_1 = -45°$ and $\theta_2 = 0°$

Given that the first modes of the PGD solution are the most informative as shown in Fig. 4.4b, we choose to illustrate the first five spatial modes. It is clear from Fig. 4.9a that the first mode is capturing the essence of the in-plane deformation (xy-plane deformation) only which is the dominant deformation, where the black square frame in the figure represents the original domain. Fig. 4.9b shows mode one from a different view, and confirms that there is no out-of-plane deformation in any way. Furthermore, mode two represents the essence of the deformation in the z-direction in a bending-like response. Note that mode two and mode four are very similar, as they share the same magnitude in opposite directions as shown in Fig. 4.9c and Fig. 4.9f. Finally, modes three and five are capturing combinations of deformations in all planes having a buckling-like response.

In order to make sense of these spatial modes and relate them to the solution showed in Fig. 4.8, we also need to look at the parametric modes presented in Fig. 4.10. First, we could easily notice that the functions are of sinosoidal nature, which is natural given that the parametric dependence of the problem only involves angles and is expressed using only cosines, sines, and their combinations. Furthermore, looking at the parametric functions at each prescribed angle, we could easily detect the influence of each parameter on the spatial modes. For instance, when we look at the first parametric function (G_1^m) in Fig. 4.10a, it is obvious that mode two (in blue) at the prescribed angle $\theta_1 = -45^\circ$ has a negative value. The negative value scales and inverts the deformation of spatial mode two, which makes sense by comparing Fig. 4.9c with Fig. 4.8. On the other hand, when we inspect the second parametric function (G_2^m) in Fig. 4.10b, we notice that mode five (in magenta) at the prescribed angle $\theta_2 = 0^\circ$ also has a negative value that scales and inverts the fifth spatial mode. As a result, and by comparing Fig. 4.9g with Fig. 4.8b, we could detect the inversion of the convexity of the structure deformation. It is important to recall that all these spatial and parametric modes are normalized; and by multiplying each spatial mode by its corresponding parametric functions and amplitude, and then summing over all the modes we recover the structural response (shown in Fig. 4.8) at the given value of the set of parameters.

Since we are aiming for finding the best fibre orientation that minimizes the Tsai-Wu criterion, it is essential to apply optimization techniques to find the optimal solution. One of the main features of PGD is that the sensitivities depend explicitly on the parameters, and therefore, the optimization problem becomes straight-forward. Consequently, for the current example where we have two parameters, it is very fast to compute the optimal solution. It is also easy to find the optimal solution by plotting the objective functions, introduced in Section 2.5, against both parameters and inspecting the obtained map for the minimum/maximum point. Note that for a higher number of parameters (more than 3), the optimization becomes much more complicated and it is also not possible to plot the objective function anymore; therefore the application of optimization strategies is inevitable. The objective functions based on the PGD solution can be easily used in combination with an optimization procedure, such as the Genetic Algorithm or a

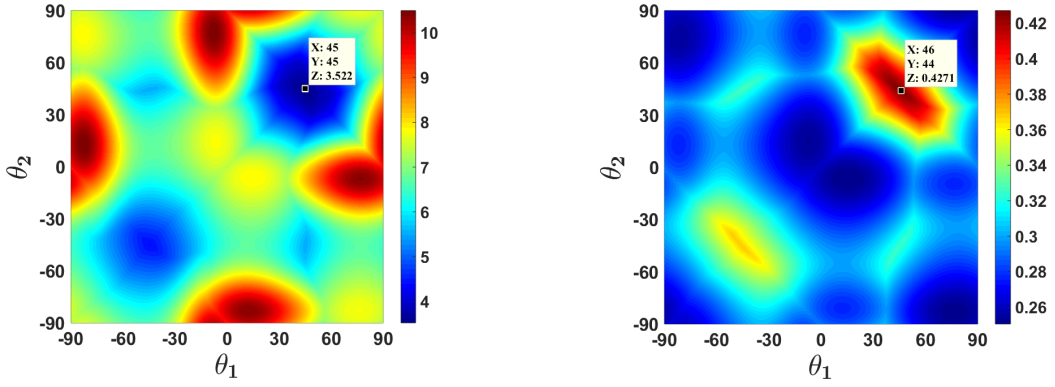


(a) Parametric function 1

(b) Parametric function 2

Figure 4.10: PGD compressed solution: Normalized parametric functions

gradient-based method, to obtain the optimal fibre orientation automatically.



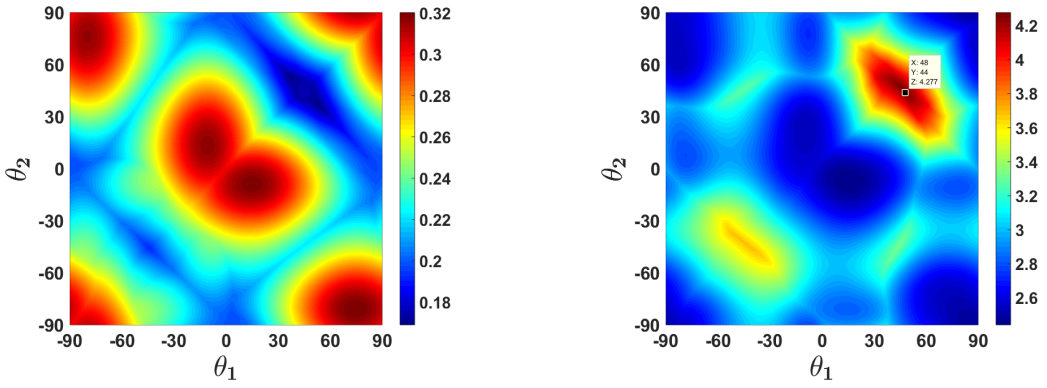
(a) Failure index ($\mathcal{I}_f$) map

(b) Safety factor index (λ_s) map

Figure 4.11: Failure criteria maps of plate under tensile load at 45°.

Fig. 4.11 shows the maps of the objective functions, namely the failure criterion and the safety factor, introduced in Section 2.4.2. Once the PGD solution has been obtained it is extremely fast to evaluate it for any value of the parameters and, therefore, one can evaluate the objective functions at every parameter value of a fine grid and produce those plots easily. The optimization in this simple case can be done by direct observation of the maps, where local and global minima/maxima are readily identified. For the example introduced in Fig. 4.3, the critical point representing the optimal value for each objective function is located at the fibre orientations $(\theta_1, \theta_2) = (45°, 45°)$. The obtained optimal solution is expected as the applied load is purely tensile in the xy-plane and is applied at 45°. Additionally, it is in agreement with the solution of the hybrid optimization using GA and a gradient-based method validating the optimization algorithms. It is easy to detect in Fig. 4.11a and in Fig. 4.11b the symmetry of the objective functions

with respect to the $(0°, 0°)$ as both parameters (layers) behave similarly with respect to the applied load. In Fig. 4.12, we show the objective functions map for the same



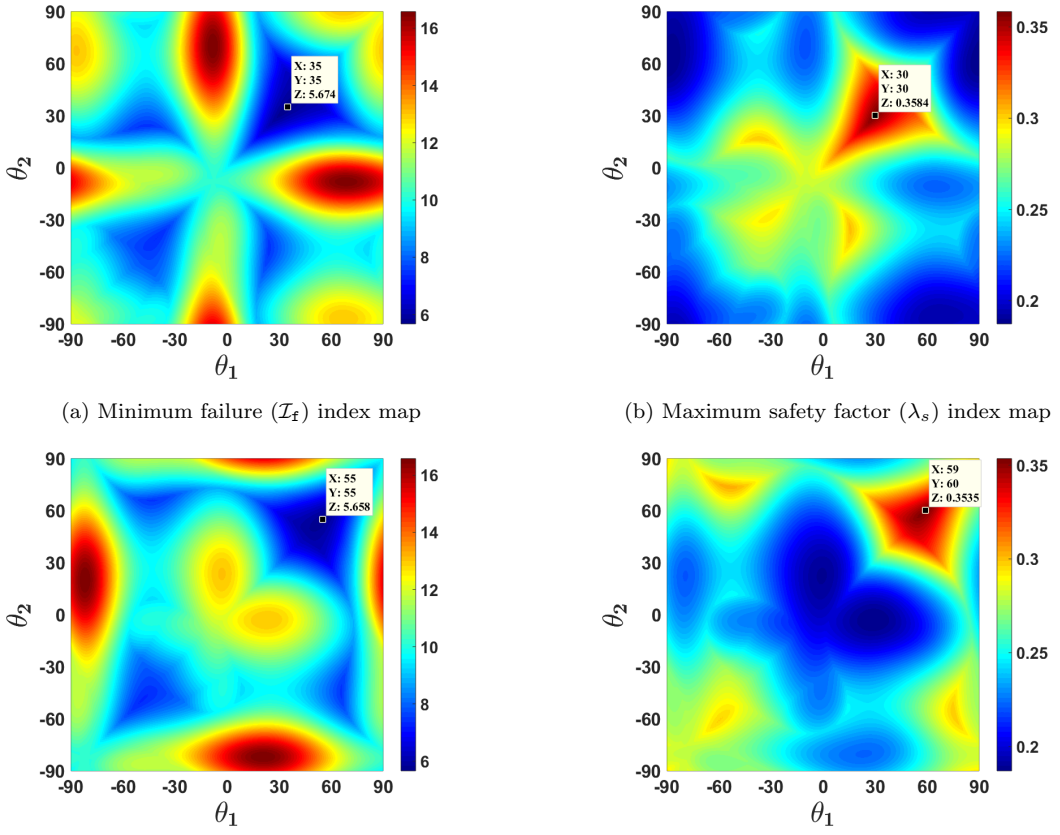
(a) Failure index ($\mathcal{I}_f$) map

(b) Safety factor index (λ_s) map

Figure 4.12: Failure criteria maps of plate under small tensile load at 45°with reduced applied load.

domain under 10% of the initial applied load in Fig. 4.11. The aim is to investigate the effect of the load variation on the optimization results. As mentioned in Section 2.4.2, the optimization of the failure criterion could be load dependent leading to inaccurate solutions which is evident when comparing Fig. 4.11a with Fig. 4.12a. On the other hand, the optimization of the safety factor does not depend on the load applied in terms of optimization results which is demonstrated in Fig. 4.12b when compared with Fig. 4.11b. Moreover, it is expected that if the angle of the applied load changes, the global minimum/maximum would change accordingly. Fig. 4.13 shows different maps corresponding to different angles of the applied load for the same domain in Fig. 4.3. We notice that the global minimum/maximum moves on the symmetric diagonal line passing through the origin with the variation of the angle of the applied load. We could also deduct that the optimization of the safety factor is more accurate and less sensitive to jumps in the objective functions.

It is important to mention that every point of these maps is equivalent to one FE solution at a particular value of the set of parameters. Note that the computing time to obtain the PGD solution is $\sim$ 2.5 hours and the map is generated in seconds. If one aims at producing the same maps based on a standard FE solution, this would require $181 \times 181 = 32761$ FE solves, that would take $\sim$ 6.5 days using the same computer power as in the PGD case. It is obvious that with the increase of parameters, the computational burden increases. As discussed in Section 3.2, the increase of parameters increases the computational complexity of the standard FE solution exponentially while it increases with PGD linearly.



(a) Minimum failure ($\mathcal{I}_f$) index map

(b) Maximum safety factor (λ_s) index map

(c) Minimum failure ($\mathcal{I}_f$) index map

(d) Maximum safety factor (λ_s) index map

Figure 4.13: Failure criteria maps of plate under tensile load at: (a) & (b) 30°, (c) & (d) 60°.

4.3 Plate with circular hole under tensile load

The second example involves a plate with a circular hole in the middle subjected to tensile load oriented parallel to the x-axis as shown in Fig. 4.14. Using the symmetry of the problem we solve only for half of the domain. This problem adds a slight ambiguity because of the existence of the hole in its geometry. It is not straightforward to predict the best fibre orientation based only on intuition, hence, we resort to optimization algorithms. The space is discretized using 390 hexahedral Serendipity elements and the discretization of each parametric dimension is the same as in the previous example in Section 4.2, that is, I_i is represented using a uniform grid of 1° spacing yielding 181 parametric nodes. Moreover, the greedy algorithm tolerance used in this section and controlling the enrichment of the PGD solution is $\xi = 10^{-4}$ unless stated otherwise. In the current section, we present three different sub-examples with different subdivisions for a plate with a hole in the centre. The plate is divided into two, four, and eight subdomains,

highlighting their effect on the optimization solution and the computational time. Note that in the current example, the subdomains are in the form of patches in the xy-plane unlike the previous example where the subdomains were represented by different unidirectional layers varying in the z-direction.

We first introduce the domain involving only two parameters (two subdomains) as shown in Fig. 4.14. The discretization of the two-dimensional parametric space is affordable as in the previous section. The plate is divided into two subdomains, each one with its independent fibre orientation in the xy-plane. Each subdivision corresponds to a different parameter $\theta_i \in I_i$ with $i = 1, 2$; where each parameter ranges in a real interval $I_i = [-90, 90]$. The material properties remain the same throughout this section and are presented in Appendix C.

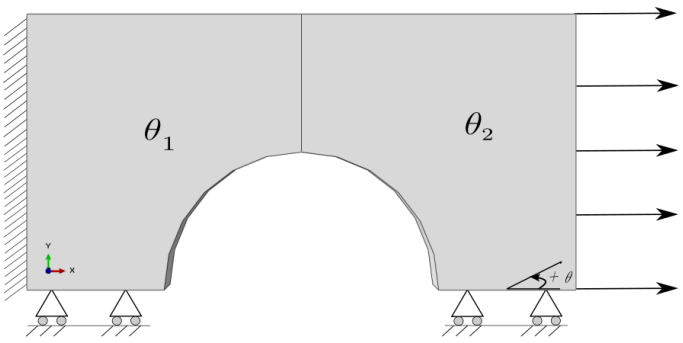
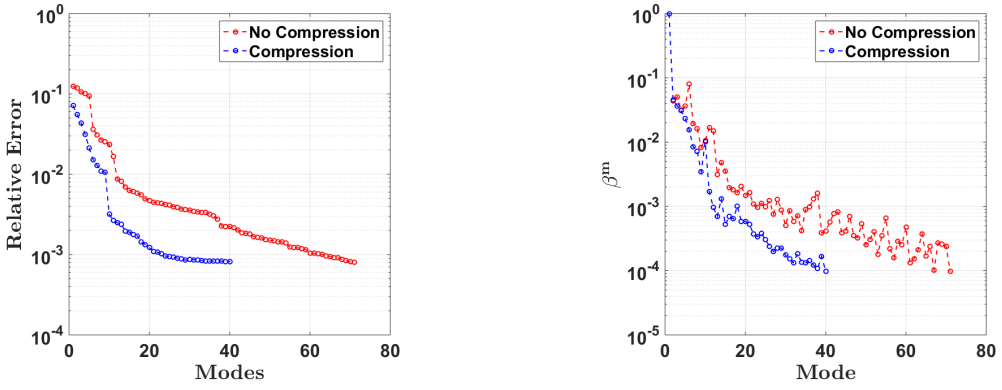


Figure 4.14: Symmetric half of a square plate with a circular hole: 2 subdomains



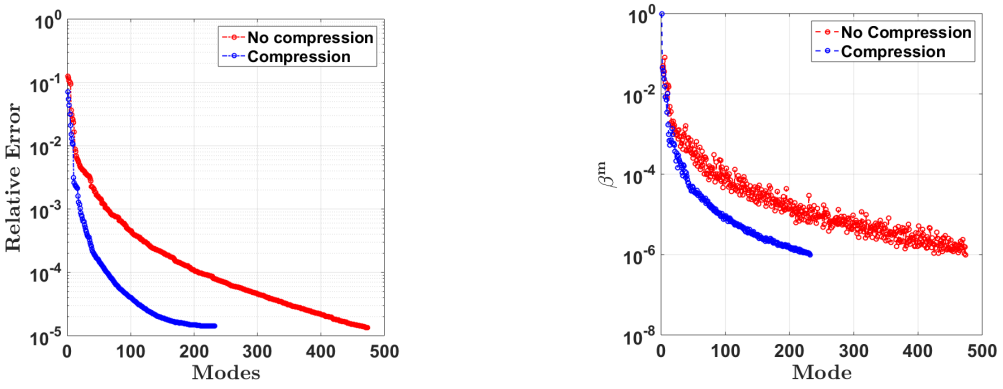
(a) Global relative error

(b) Amplitude evolution

Figure 4.15: Plate with hole under tensile load at 0° with two parameters for a greedy tolerance $\xi = 10^{-4}$

Similar to the example in the previous section, Section 4.2, we first compute the standard FE solution at different values of the set of parameters; and then compute the error

between the PGD and standard FE using Eq. (4.2). Results of the PGD parametric solution show a great agreement with those obtained using the standard FE with errors around $\sim 0.1\%$ with 40 modes in the PGD compressed solution. The error is estimated using subsets of the parametric grid with 3° angle step as shown in the previous example in (4.2). In Fig. 4.15a, the convergence curve of the compressed solution shows a sharp decrease of the error in the first 20 modes before the curve starts to reach a *plateau* around 10^{-3}. We could also deduce that the PGD-compression is performing very well because its error (blue curve in Fig. 4.15a) is reaching the same value as the PGD uncompressed solution error (red curve in Fig. 4.15a). In Fig. 4.15b, we observe the evolution of the amplitudes of the PGD solution. The PGD-compression routine yields a reduction of 43.5% in the total number of modes.



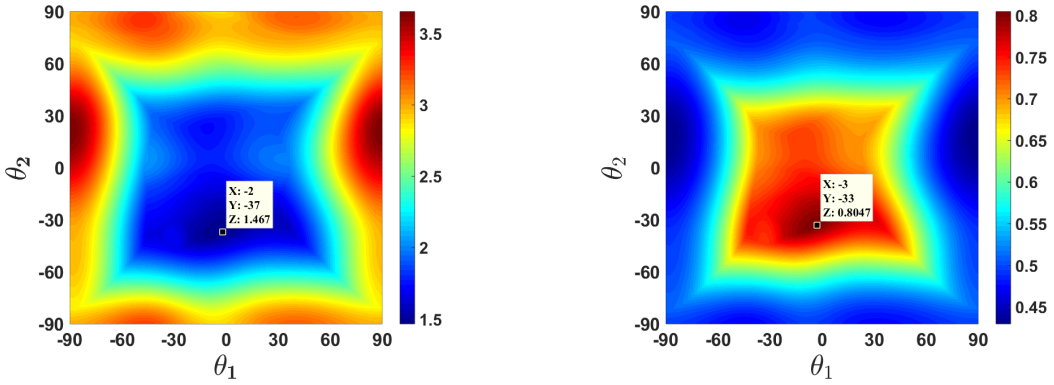
(a) Global relative error

(b) Amplitude evolution

Figure 4.16: Plate with hole under tensile load at 0° with two parameters for a greedy tolerance $\xi = 10^{-6}$

In Fig. 4.16 we could see the evolution of the error and the amplitudes with a stricter greedy algorithm tolerance leading to a lower error value between the PGD and standard FE solutions ($\sim 0.001\%$) with the price of a bigger number of modes. The optimization results are presented in the maps of the objective functions shown in Fig. 4.17a and Fig. 4.17b with optimal values $(\theta_1, \theta_2) \approx (-3°, -33°)$. The results give a general description of the stress distribution in the structure through the failure index and the safety factor. We expect a stress concentration around the hole and, therefore, we expect to see the fibres encircling the hole to minimize failure. Consequently, increasing the number of parameters would result in more subdomains that capture better the variation of stresses around the hole and thus leading to better optimization results.

The second example in the current section involves four parameters as shown in Fig. 4.18. With the aim of solving the parametric PDE to cover the whole space of solutions, the discretization of the four-dimensional parametric spaces becomes impractically expensive



(a) Failure index $\mathcal{I}_f$ map

(b) Safety factor λ_s map

Figure 4.17: Plate with hole under tensile load at 0° with two parameters: Objective functions maps

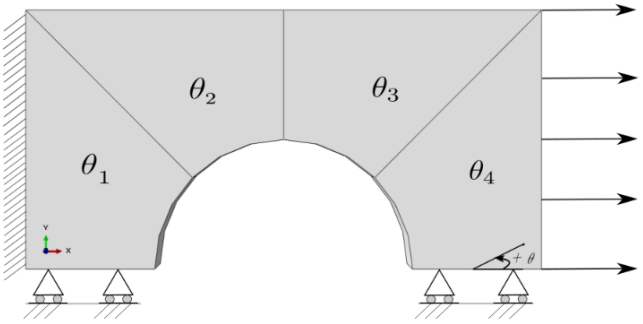


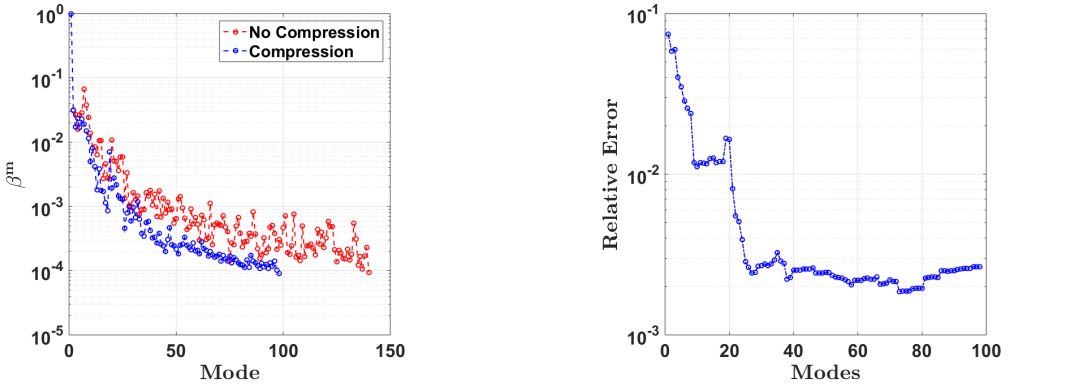
Figure 4.18: Symmetric half of a square plate with a circular hole

if standard techniques are applied because the number of parametric points increases exponentially with the number of dimensions (shown in Fig. 4.24). The separable character of the PGD solution, on the other hand, makes this problem tractable as every dimension is discretized independently.

The plate is divided into four sub-domains, each one with its independent fibre orientation determined by the corresponding parameter $\theta_i \in I_i$ with $i = 1 \dots 4$; where the range of all parameters θ_i, is $I_i = [-90°, 90°]$ yielding 181 parametric nodes. Note that the discretization of the coupled four-dimensional parametric space would require $181^4 > 10^9$ points, whereas the separated representation requires 181×4 points. The parametric part of the solution is stored in $181 \times 4 \times m$ points, being m the number of terms used in the solution (98 for the compressed solution for this example).

The parametric PGD solution has been computed with the same tolerances as in the previous example and, after compression, the solution has 98 modes. Amplitudes of the

modes are shown in Fig. 4.19a. The PGD-compression yields a reduction in the number of modes of 30%. Furthermore, having four parameters makes integrating the error in the parametric domain too expensive as the number of FE solves is enormous (181^4). As a reference, in Fig. 4.19b we provide a convergence curve of the local error in space for one given point in the parametric space, $\boldsymbol{\theta}^{\mathtt{Opt}}$, that happens to be the optimal solution found as described next.



(a) Amplitude evolution for $\xi = 10^{-4}$

(b) Relative error measured at optimal solution $\boldsymbol{\theta}^{\mathtt{Opt}}$

Figure 4.19: Amplitude of the PGD modes and the error in space measured in one parametric point

In this four-dimensional case it is not possible to visualize the objective functions and, hence, finding the critical point of the failure criterion by inspection is not feasible; and thus an optimization algorithm must be employed.

As we have seen in the previous example, the objective functions defined by the failure criterion and the safety factor are not convex and local minima/maxima are present. Applying a gradient-based method converges often very fast however there is always the risk to be stuck in a local minimum/maximum if the initial guess is not close enough to the optimal global solution. On the other hand, applying a global method such as the Genetic Algorithm, leads to a global optimum with the price of a slow rate of convergence. Therefore we obtain the optimal solution using a hybrid strategy performed in two steps with the aim of minimizing the computation time. First we use the Genetic Algorithm to perform a global minimization/maximization without large accuracy. Second, we use a gradient-based method to reach the global minimum/maximum starting from the solution of the first step. Both optimization methods are implemented using general built-in Matlab functions (`ga` and `fmincon` functions). In the following, we focus more on the optimization of the objective function involving the safety factor (λ_s) as it yields more accurate results as discussed in Section 2.3. However, we will show some results regarding the failure index ($\mathcal{I}_{\mathtt{f}}$) for comparison purposes.

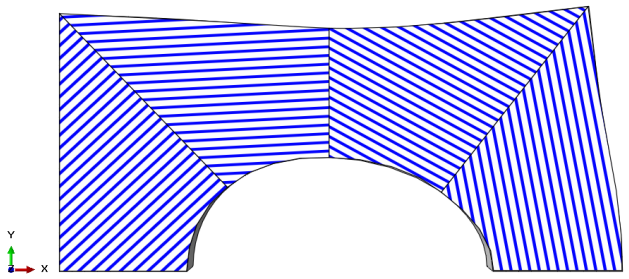


Figure 4.20: Optimal fibre orientations on the deformed domain obtained by a global maximization of the safety factor λ_s using first the Genetic Algorithm and then a gradient-based method.

The optimal fibre orientations are shown in Fig. 4.20 and presented in Table 4.1. The result agrees well with the intuition that the fibres should encircle the hole where there is stress concentration. Note that, the output angles of the optimization in Table 4.1 for the Genetic Algorithm are integer numbers because the population of angles in the input are integers as well. The number of evaluations of the objective function in this optimization problem using the GA is approximately 100,000 out of a total around 1×10^9 possible evaluations. Using this result as the initial guess for the gradient method, we obtain a very robust and precise solution in an efficient way.

Optimal Angles				
	`ga` function		`fmincon` function	
θ_1	-73°	42°	-73.0371°	42.062°
θ_2	4°	3°	3.9244°	2.9944°
θ_3	-26°	-22°	-26.0131°	-22.4586°
θ_4	-88°	-83°	-87.9362°	-84.1544°
Index value	$\min(\mathcal{I}_\mathtt{f}) = 1.4237$	$\max(\lambda_s) = 0.8254$	$\min(\mathcal{I}_\mathtt{f}) = 1.4235$	$\max(\lambda_s) = 0.8255$
CPU time	~ 2 min	~ 40 min	~ 0.2 min	~ 1 min

Table 4.1: Optimized angles for square plate with circular hole for different objective functions.

We could deduce from Table 4.1 that the GA algorithm performs very well with a relatively small number of evaluations in this particular example that we are solving. The GA predicts efficiently the optimal solution very close to the global optimum for both objective functions. The obtained solution is then a perfect choice as an initial guess for the gradient-based method.
In Table 4.2, we could see the difference between the optimization of both the failure

index and the safety factor when applying different load values. It is easy to notice that the optimization results of the failure criterion are load dependent while the safety factor results are consistent regardless of the load magnitude (as discussed in Section 2.3).

Optimal angles for different applied loads values				
	1250 kN		5000 kN	
θ_1	-73°	42°	83°	43°
θ_2	4°	3°	6°	3°
θ_3	-26°	-22°	-39°	-20°
θ_4	-88°	-83°	-80°	-85°
Index value	$\min(\mathcal{I}_\mathtt{f}) = 1.4235$	$\max(\lambda_s) = 0.8255$	$\min(\mathcal{I}_\mathtt{f}) = 18.9261$	$\max(\lambda_s) = 0.264$

Table 4.2: Optimized angles for square plate with circular hole with different applied load values.

In a nutshell, once the parametric solution is computed by PGD, its evaluation for any parametric point is extremely fast and, therefore, it is possible to perform a very large number of evaluations of the objective function within the optimization scheme. Unfortunately in some cases, where the problem is very sensitive to the initial guess when using a gradient-based method, we resort only to the Genetic Algorithm to ensure accuracy which leads to more computational efforts.

Optimal angles maximizing the safety factor λ_s using GA and a gradient-based method								
# of GA evaluations	100		1000		10,000		1,000,000	
Opt. Function	`ga`	`fmincon`	`ga`	`fmincon`	`ga`	`fmincon`	`ga`	`fmincon`
θ_1	-79°	-59.024°	-58°	-58.562°	-57°	-57.176°	42°	42.230°
θ_2	35°	3.522°	3°	2.775°	2°	1.871°	3°	2.888°
θ_3	31°	-21.852°	-19°	-21.143°	-22°	-22.012°	-21°	-21.054°
θ_4	-15°	-79.962°	-69°	-85.388°	-84°	-83.835°	-84°	-84.949°
$\max(\lambda_s)$	0.6847	0.8235	0.8199	0.8237	0.8236	0.8236	0.8254	0.8255
CPU time	~ 0.1 min	~ 2 min	~ 0.6 min	~ 0.5 min	~ 4.2 min	~ 0.2 min	~ 450 min	~ 0.2 min

Table 4.3: Different GA precision yielding values used as initial guesses for the gradient-based method

In Table 4.3, we present the maximized optimal solutions obtained using different numbers of evaluations of the safety factor λ_s with the aid of the Genetic Algorithm. The

gradient-based solutions having the GA solutions as their initial guesses are also presented. It is evident that for this particular problem we reach the global solution with 100,000 evaluations of the objective function as shown earlier in Table 4.1.

This is confirmed when we converge to the same solution even after increasing the precision of the GA by employing 10^6 evaluations as shown in Table 4.3. In the case with high precision, the use of a gradient-based method with the GA solution as an input becomes a complementary step to ensure convergence. On the other hand, when applying the GA with low precision (i.e. small number of evaluations) and then introduce the solution to the gradient-based method as initial guess, we diverge from the global optimal solution due to the existence of local maxima.

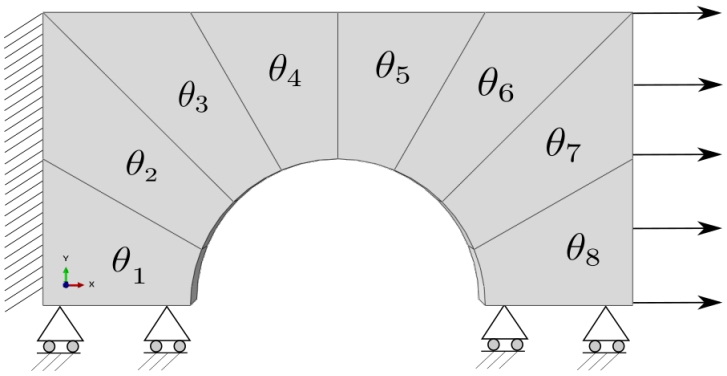
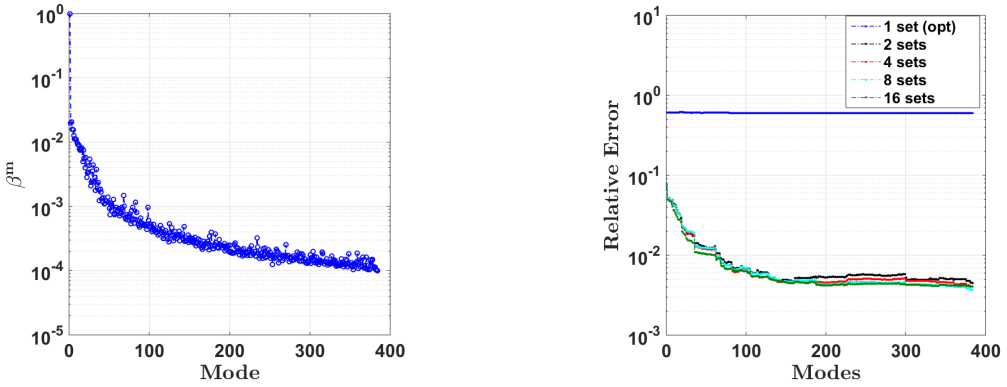


Figure 4.21: Symmetric half of a square plate with a circular hole subdivided in 8 patches

The third example of the current section involves eight parameters shown in Fig. 4.21. The plate is divided into eight sub-domains, each one with its independent fibre orientation determined by the corresponding parameter $\theta_i \in I_i$ with $i = 1 \dots 8$; where the range of all parameters θ_i, is $I_i = [-90°, 90°]$ yielding 181 parametric nodes.

Note that the discretization of the coupled eight-dimensional parametric space would require $181^8 > 10^{18}$ points, whereas the separated representation requires 181×8 points. The parametric part of the solution is stored in $181 \times 8 \times m$ points, being m the number of terms used in the solution (384 for the compressed solution for this example).

The parametric PGD solution has been computed with the same tolerances as in the previous examples and, after compression, the solution has 384 modes. The amplitudes of the modes of the compressed solution are shown in Fig. 4.22a. We could easily observe that the number of modes needed for convergence increased with the number of parameters as expected. Furthermore, having eight parameters makes integrating the error in the parametric domain impossible as the number of standard FE solves for the full parametric space is astronomical (181^8). As a reference, in Fig. 4.22b we provide a convergence curve of the error in space for different subsets of the parametric space which are chosen randomly; except for the error measured at one set (the blue curve)



(a) Amplitude evolution for $\xi = 10^{-4}$

(b) Relative error measured at different subsets

Figure 4.22: Amplitude of the PGD compressed solution and the error in space measured in one parametric point, $\boldsymbol{\theta}^{\mathtt{Opt}}$, defined inTable 4.4

that happens to be the optimal solution found as shown in Table 4.4. We could notice that the error is relatively high for the optimal set compared to the previous examples with two and four parameters. This is expected as the parametric space is much larger and therefore the percentage of contribution of one parametric point to the global error is much lower. Moreover, the more we include different parametric points in the global error evaluation, the more the error decays. The optimal fibre orientations are

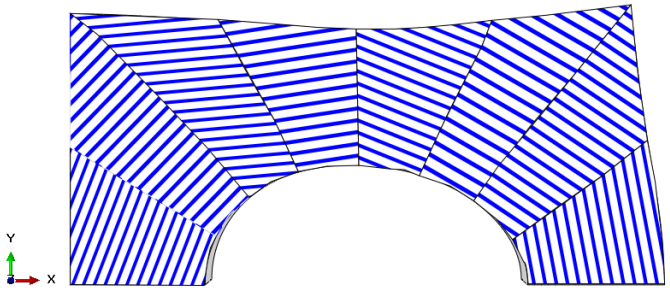


Figure 4.23: Optimal fibre orientation on deformed domain

shown in Fig. 4.23 and presented in Table 4.4. The results agree well with the previous example, with the intuition that the fibres should be oriented around the hole. The number of evaluations of the objective function in this optimization problem using the GA is approximately 100,000 out of a total around 1×10^{18} possible evaluations. As the parametric space is very large, using a low precision GA and then the gradient-based method does not give a good solution. Therefore, we only use the GA algorithm with

high precision in this example.

Optimized angles of the safety factor using GA				
# of GA evaluations	1000	10,000	100,000	1,000,000
θ_1	81°	86°	70°	84°
θ_2	42°	43°	42°	41°
θ_3	5°	6°	6°	4°
θ_4	−6°	8°	8°	−20°
θ_5	−20°	−21°	−23°	17°
θ_6	−24°	−25°	−26°	−51°
θ_7	−26°	−25°	−30°	−61°
θ_8	45°	−86°	−85°	−82°
$\max(\lambda_s)$	0.8249	0.8803	0.879	0.8501
CPU time	∼ 1.2 min	∼ 12 min	∼ 120 min	∼ 1300 min

Table 4.4: Different number of evaluations yielding different GA precision

The optimization problem is naturally evaluating the forward problem numerous times. The great advantage of PGD is that, through the computational vademecum, the whole parametric space is available and browsing it for any value of the set of parameters is very fast. Generally speaking, mesh based techniques are more accurate than PGD, however very expensive in a multi-query application, like optimization, especially when the number parameters is large. Fig. 4.24 shows the CPU time needed for the PGD and standard FE to explore the whole parametric space; and for standard FE to explore a reduced parametric space (30% of the full parametric space). We could deduce from the trend of the graph that the PGD is by far computationally cheaper than standard FE when considering a number of parameters more than two.

We have already noticed in the current chapter that altering the domain sub-divisions to assign different parameters changes the optimal solution. Since the choice of sub-divisions is done manually based on intuition, therefore, in the next chapter we would like to explore a little bit further a way to automatically choose the sub-divisions based on a mechanical measure and see if we yield better optimal results.

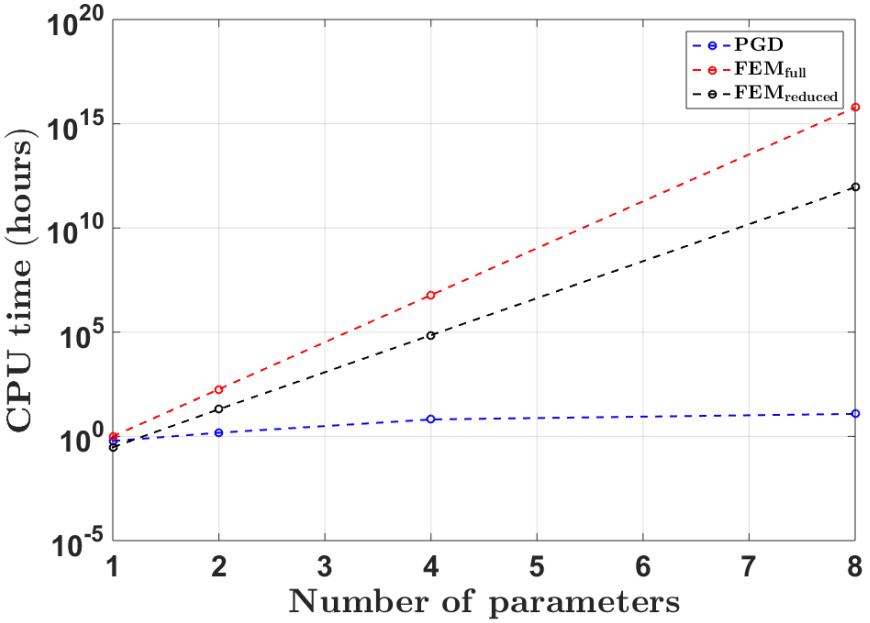


Figure 4.24: CPU time evolution of PGD vs standard FEM

Chapter 5

DOMAIN DECOMPOSITION USING DATA ANALYSIS TECHNIQUES FOR EFFICIENT PGD PARAMETERIZATION

5.1 Preliminaries

Up until this point we have shown how the PGD framework obtains a generalized solution for composite laminates parameterized with fibre orientations. To obtain the generalized solution, it is necessary to parameterize it with the angles describing the fibres (that are eventually optimized) and assign each angle to a predefined independent subdomain (e.g. the 4 subdomains in Figure 4.18). Once the generalized solution is obtained and stored, through the computational vademecum we could optimize the fibre orientations in the initially chosen independent partitions (e.g. Figure 4.20). One possible drawback of the methodology is that when we exceed a certain number of parameters ($\sim$ 15 subdomains), the PGD convergence in an acceptable time is not guaranteed. Thus, one should pay attention on how to choose the partitions of the domain to efficiently parameterize the problem. One possible way to smartly partition the domain is to group elements having similar mechanical features. The partitioning occurs in a way such that each parameter (i.e. the fibre orientation) is constant inside each subdomain where it is assigned. Thus, a natural idea is to select the parameters such that the stress state tends to be also as

uniform as possible in each subdomain. In order to identify zones where the stresses are typically uniform, a large number of solutions of different configurations of the system is analyzed, aiming at identifying the zones in which the variability of the stresses is least; these zones are detected using clustering techniques. Once detected, the parametric problem supported by this partitioning of the domain is solved with a PGD strategy and the optimal values of the parameters are readily identified. Compared with the arbitrary, regular domain partitioning (shown in Chapter 4), these new "smart" partitions result in designs improving the objective functions (failure index), and therefore improving the overall methodology.

Clustering is a fundamental data analysis tool that could be classified as an unsupervised learning method. Clustering aims to automatically group elements having similar features in a data set into coherent groups or clusters. Clustering is applied in many fields such as:

- Medicine in image segmentation, differentiation between different types of tissue [81, 82].
- Biology and bioinformatics [83].
- Business and marketing: in market research to group people with same taste or needs for example [84].
- Climatology: to detect weather patterns [85].

There are many types of algorithms that perform clustering. These algorithms are different in their notion of what qualifies as a cluster and how to find it. One main classification is hard and soft clustering. Hard clustering is when an element in a cluster cannot belong to two or more clusters. Soft clustering is when an element could be assigned to different clusters like C-means algorithm [86]. We will focus in this thesis on hard clustering algorithms such as K-means and hierarchical clustering. For the sake of completeness, the K-means and the hierarchical clustering algorithms are explained in detail in Appendix D.

The road-map of the current chapter is presented in Fig. 5.1.

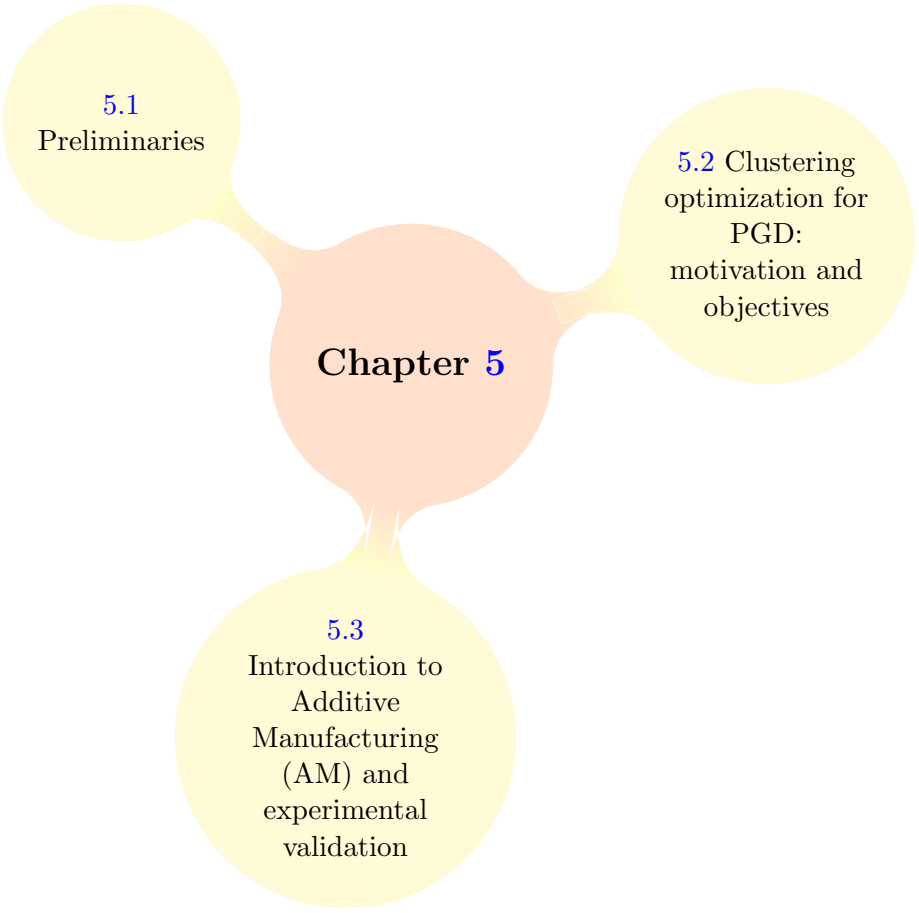
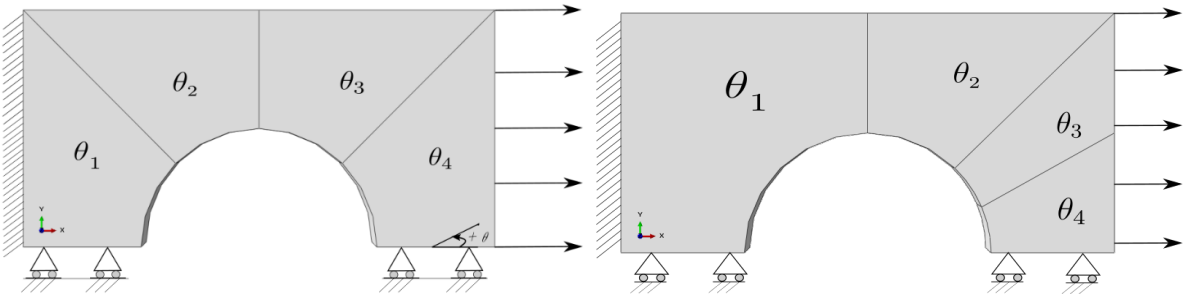


Figure 5.1: Structure of Chapter 5.

5.2 Clustering optimization for PGD: motivation and objectives

In the previous chapter we showed the effect of increasing the number of parameters on the optimal fibre orientation solution. It was shown that the more we increase the subdivisions (i.e. parameters), the more we represent the problem more accurately and obtain better results in terms of the safety factor (or the failure index). A simple example in Fig. 5.2 shows two different partitioning strategies, and it was found that even when we partition the domain in a different way while maintaining the number of subdivisions fixed, we obtain different optimal results; with an increase in the safety factor by $\sim 25\%$.

Furthermore, when we increased the number of partitions from four (as in Fig. 5.2a) to



(a) symmetrically partitioned domain (b) Shifted partitioned domain

Figure 5.2: Example of two partitioned domain with four sub-divisions

eight subdivisions (as in Fig. 4.21) while maintaining the same symmetrical partitioning strategy, there was also an improvement in the safety factor by $\sim 7\%$. We could easily deduce that both changing the partitioning patterns and increasing the number of partitions affects the safety factor index and, consequently, affects the optimal fibre orientation results. Since the PGD convergence becomes slow after a certain number of parameters ($\sim$ 15 parameters), therefore, we are interested in maximizing the usage of the "limited" number of possible subdomains (i.e. parameters). Moreover, in the previous chapter, the choice of the subdomains was always based on intuition and it is not necessarily the most descriptive partitioning of the domain especially when the structure under investigation has a complex stress distribution and it is difficult to predict it. As a result and in an attempt to have more descriptive subdivisions, we perform an analysis of the domain prior to parametrizing and solving the problem using PGD.

The preanalysis consists of collecting as much data as possible, e.g. the stress components, by taking snapshots of the system in different fibre orientations. Note that each element of the FE mesh could have an independent fibre orientation in the preanalysis when taking the snapshots in order to have a more descriptive data. We then reduce the dimensionality of the collected data using Principal Component Analysis (PCA), and we also obtain the most important uncorrelated modes representing the data. Finally we perform some clustering analysis and clustering optimization using the *factors* (or modes) obtained from PCA that are computed based on the collected data. For our problem, we collect the stress components and the safety factor index (λ_s) for each element for different snapshots of the system. All these steps would be considered as a pre-process, before solving the mechanical and the optimization problems using PGD.

Inspired by the work of Alaimo *et al.* [22], we could explain the clustering optimization process in four main steps as follows:

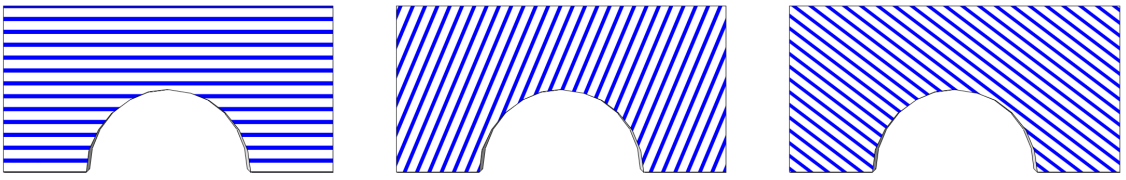
(1) **Preanalyses:** snapshots of the system's stresses and safety factor index evaluated at each finite element for different orientations are taken and stored.

(2) **Principal Component Analysis:** responsible for the data transformation from correlated fields to uncorrelated new components which describe the problem in a lower dimension.
(3) **Clustering of factors and their intersection:** the clustering techniques are applied to the factors (components) obtained from PCA.
(4) **Error computation and clustering optimization:** clustering optimization in order to find the best clusters representing the data.

For the sake of clarity and completeness, we will go through each point in detail.

5.2.1 Preanalyses

The main goal of the preanalyses step is to calculate quantities that represent the mechanical response of the system in order to decompose the domain into groups or patches of elements based on mechanical measures. We will build our data set by pre-computing the stresses and the safety factor index in each finite element for a given set of different fibre orientations using standard FE. It is important to note that the more we collect data, the more we will have accuracy in our results but with the price of increasing the computational cost. It is possible to assume that each finite element is independent from the others in terms of fibre orientation in a single snapshot to have a better description of the problem, however, this yields computationally very expensive preanalyses. Since, we would like to minimize the computational burden in the preanalyses step, and we aim only at having a general description of the domain, we will assume that all the finite elements have the same orientation in a single snapshot as shown in Fig. 5.3.

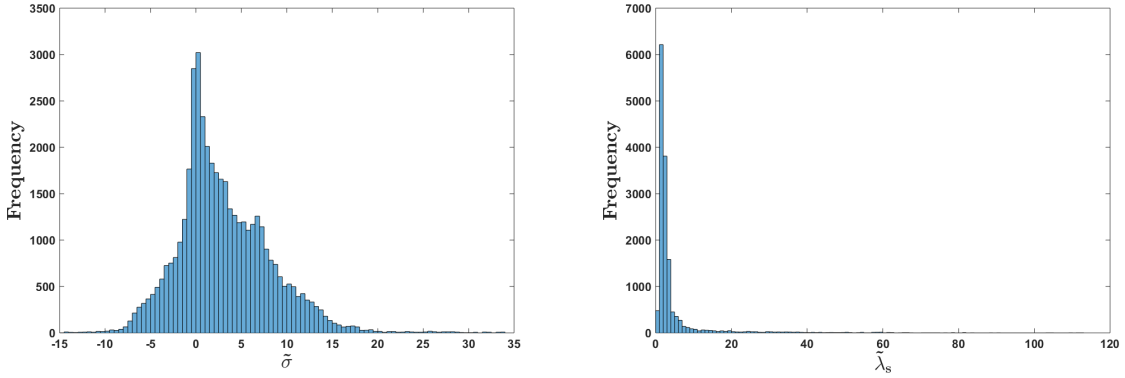


(a) Fibres at 0° (b) Fibres at 70° (c) Fibre at −45°

Figure 5.3: Snapshots of different fibre orientations in the domain

The idea now is to solve the system at different values of the fibre orientations and the stresses and safety factor evaluated at the centre of each element are then stored. In this manner we end up with matrices or vectors of data for each snapshot (i.e. each fibre orientation) having the size of $\mathtt{n_{el}} \times 6$ for the stresses and $\mathtt{n_{el}} \times 1$ for the safety factor where $\mathtt{n_{el}}$ is the total number of elements in our FE mesh. The six components of the stresses are reduced to three by taking only the in-plane components to minimize the

computational burden. We could then arrange the data in two different data matrices that we will denote as $\tilde{\boldsymbol{\sigma}}$ and $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}}$ for stresses and the safety factor respectively. Each of the raw data matrices is composed of $\mathtt{n_{el}}$ rows and $\mathtt{N_c}$ columns. For example if we take 50 snapshots of the system, then the size of the stresses data matrix will be $\mathtt{n_{el}} \times \mathtt{N_c}$ where $\mathtt{N_c} = 150$ because for each snapshot we store 3 stress components yielding $\mathtt{N_c} = 3 \times 50$ columns. Whereas for the safety factor data matrix will be $\mathtt{n_{el}} \times 50$ as each snapshot we store one vector containing the safety factor values at the centre of each element. It is argued in [87] that the K-means algorithm, which we will apply in our analyses, is sensitive to the skewness of the data and could give inaccurate or even wrong results. It is then important to verify first if the data collected is balanced or not before performing any kind of analysis on it. We resort to histograms plots where we show the distribution of the data. It turns out that the stress raw data distribution is balanced while the distribution of the safety factor raw data is very skewed as shown in Fig. 5.4. The



(a) Histogram of stress raw data $\tilde{\boldsymbol{\sigma}}$

(b) Histogram of safety factor raw data $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}}$

Figure 5.4: Raw data distribution histograms

common practice and the easiest way to overcome this issue is to transform the safety factor raw data. There are many ways to do transformation of data that could be found in the literature. We will adopt one of the most popular transformations that is called the power transformation, and namely, we will apply the logarithmic operator on the data. After applying the transformation, the data becomes more balanced as shown in Fig. 5.5. Once we have obtained the data matrices, we could possibly apply PCA to them in order to convert the correlated quantities to uncorrelated ones. By doing so, we ensure that we could remove any redundancy in the data and only work with the most informative unique components as we will see in the next subsection.

5.2.2 Principal Component Analysis

PCA is one of the oldest and best known of the techniques of multivariate analysis [88]. The idea behind PCA is to reduce the dimensionality of the data set having correlated

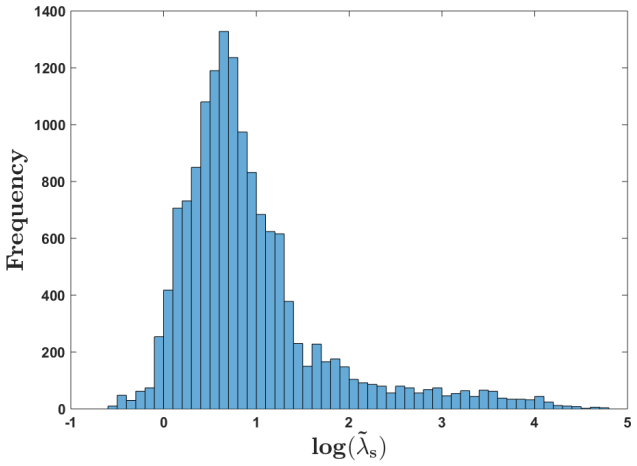
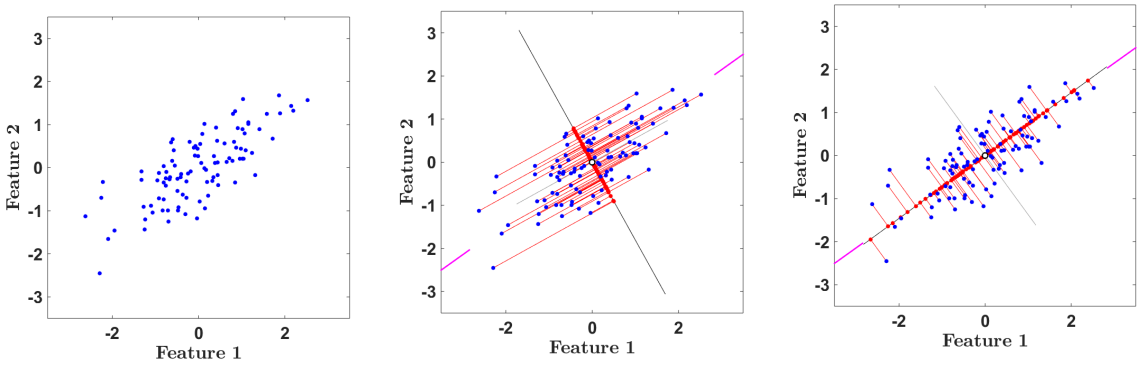


Figure 5.5: Histogram of the transformed safety factor raw data $\log(\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}})$

variables while attempting to keep its variance as much as possible. The dimensionality reduction is necessary as it certainly saves up computer memory and speeds up our algorithms. The reduction of the data using PCA produces a new set of variables, called factors or principal components, that are uncorrelated. The most important information (the most variation of all the data) is held in the first few components [88].

In Fig. 5.6a is an example of a data set (blue dots) having only two features. What PCA does is that it approximates the original data set by projecting it on a new line, that is the black line passing through the data in Fig. 5.6b. PCA tries to find the best line that minimizes the projection error, i.e. the sum of squared distance (the red lines) between the data (blue dots) and the newly projected data (red dots), while maintaining as much as possible the variance of the data. Note that there is an equivalence between deriving the principal components by minimizing the projection error or by maximizing the data variance [89]. We can notice that in Fig. 5.6b is not a good fit as the variance of data is low and the projection error between the blue dots and the black line is maximal. Whereas in Fig. 5.6c, the data is very well spread, i.e. having high variance, and the minimum projection error is reached. In this way we have reduced the dimensionality of the data by representing two features with only one line (2D to 1D).

We could look at PCA as solving an eigenvalue-eigenvector problem, where we try to find the direction (eigenvector) that maximizes the variance of the projected data. The eigenvector maximizing the variance possesses the largest eigenvalue. In our case, we particularly use Singular Value Decomposition (SVD) to find the eigenvectors and eigenvalues, as it was reported to be more stable numerically [88]. We could formulate then



(a) Data set example (b) Data projection with low variance (c) Data projection with high variance

Figure 5.6: PCA example between two features

the problem in a general way as follows:

$$\boldsymbol{\Sigma} = \frac{1}{m}\boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}, \tag{5.1}$$

where $\boldsymbol{X}$ is a $m \times \mathtt{N_c}$ generic data matrix that is equivalent to $\tilde{\boldsymbol{\sigma}}$ or $\log(\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}})$ in our problem, m is the total number of observations in the generic data matrix ($\mathtt{n_{el}}$ in our problem), and $\boldsymbol{\Sigma}$ is the covariance matrix.

Remark 5.1. Normally a generic raw data matrix $\boldsymbol{X}$ having features with different scales should be normalized or standardized depending on the data distribution. It is a normal pre-processing procedure and a common practice when applying PCA. However, in our case, we use the raw data because we would like to keep the variability of the data as it is and also because the data have the same scale and units.

Using the covariance matrix obtained from (5.1), we obtain the eigenvalues λ_i and the corresponding eigenvectors $\boldsymbol{v}^i$, with $i = 1, \ldots, \mathtt{N_c}$, which we then use to determine our principal components (or factors). It is important to note that in order to have a well-defined problem, this relation must hold $\boldsymbol{v}^{\mathsf{T}}\boldsymbol{v} = 1$. Now we could define the factors (principal components) as follow,

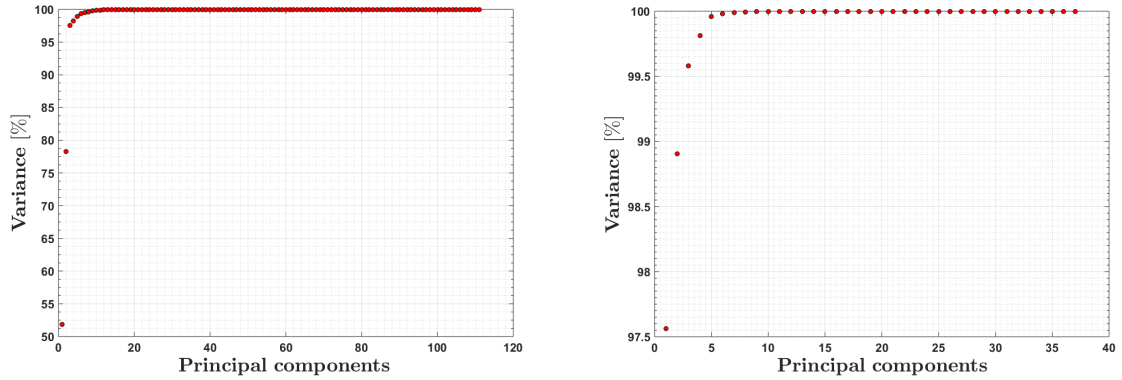
$$\mathbf{f}^i = \boldsymbol{X}\boldsymbol{v}^i, \tag{5.2}$$

with the eigenvalues expressed in terms of the obtained factors as follows,

$$\lambda_i = \frac{1}{m}\sum_{j=1}^{m}(\mathrm{f}_j^i - \hat{\mathrm{f}}^i)^2, \tag{5.3}$$

where $\hat{\mathrm{f}}^i$ is the mean of $\mathbf{f}^i$. The eigenvalues obtained from PCA are normally ordered in descending order (i.e. $\lambda_1 > \lambda_2 > \ldots, \lambda_{\mathtt{N_c}}$). Consequently, the first factors have more importance by having the most variance of the data (see proof in Appendix E). Furthermore, it is worth mentioning that the eigenvectors are orthogonal which ensures

there is no redundancy of data between the factors obtained. The evolution of the cumulative variance against the number of factors computed for the stresses, $\tilde{\boldsymbol{\sigma}}$, and the transformed safety factor data, $\log(\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}})$, is illustrated in Fig. 5.7.



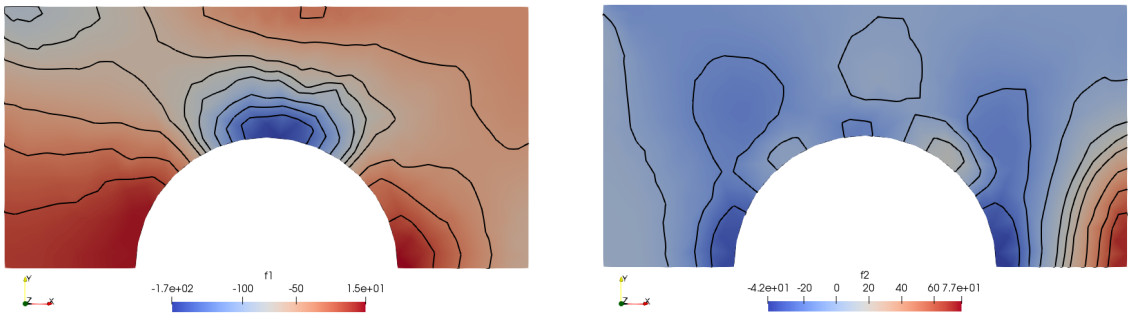
(a) Stress based curve

(b) Transformed safety factor based curve

Figure 5.7: Evolution of the cumulative variance with the number of factors

Fig. 5.7 is very important as it shows us the percentage of the total variance that we get when we accumulate the information from the factors and accordingly we decide how many factors to consider in our analysis. It is shown that three factors cover more than 95% of the cumulative variance of the data while four factors cover more than 97% of the data.

In our analysis in the following sub-sections, we choose to use four factors as they cover a satisfactory percentage of the total cumulative variance of the raw data leading to both accuracy and computational efficiency. In Fig. 5.8 we show the first four factors obtained from applying PCA, according to Eq. (5.2), on the stress raw data, $\tilde{\boldsymbol{\sigma}}$. Similarly, in Fig. 5.9 we show the first four factors obtained from applying PCA on the transformed safety factor raw data, $\log(\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}})$. For completeness, we will also show the factors obtained from applying PCA on the safety factor data before transformation and its clustering in Appendix F.



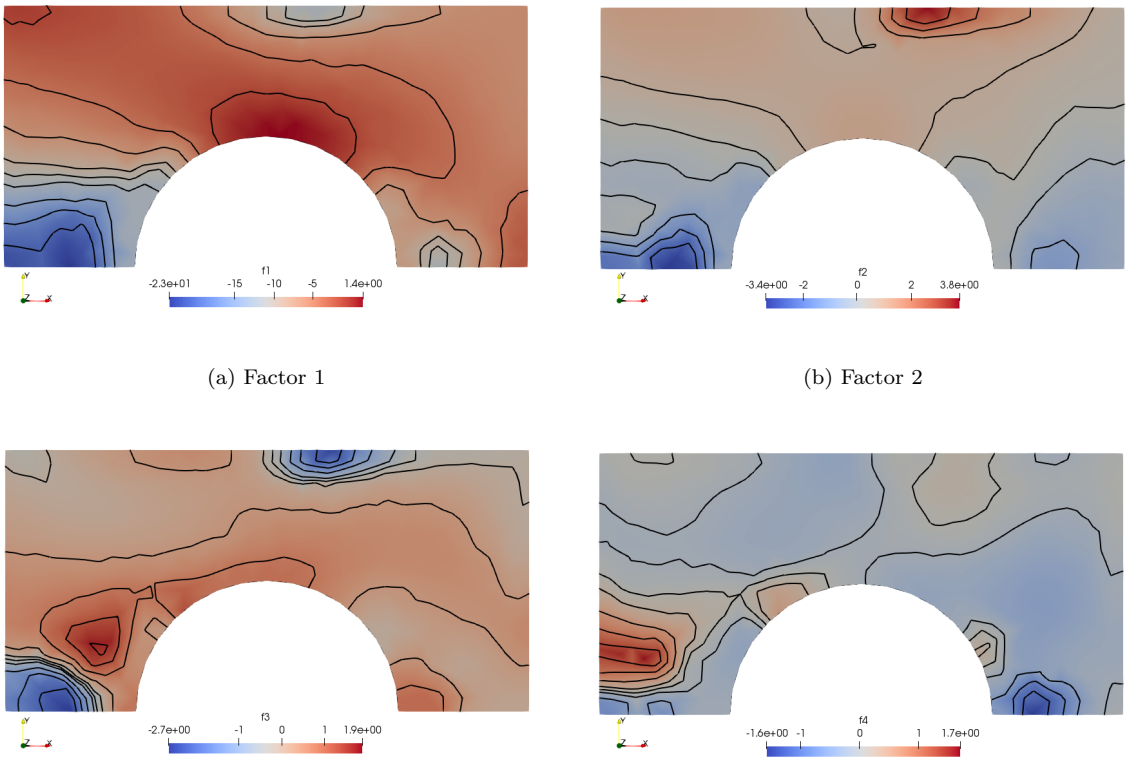
(a) Factor 1

(b) Factor 2

(c) Factor 3

(d) Factor 4

Figure 5.8: The first four factors obtained from applying PCA on stress raw data



(a) Factor 1

(b) Factor 2

(c) Factor 3

(d) Factor 4

Figure 5.9: The first four factors obtained from applying PCA on the transformed safety factor raw data matrix $\log(\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}})$

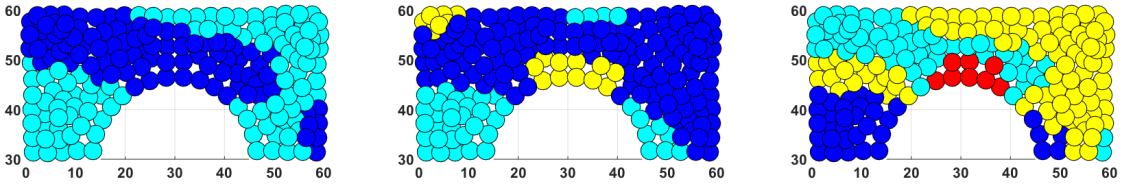
In the next sub-section we show how the obtained factors are divided into sub-clusters using the clustering techniques introduced earlier in this chapter.

5.2.3 Clustering of factors and their intersection

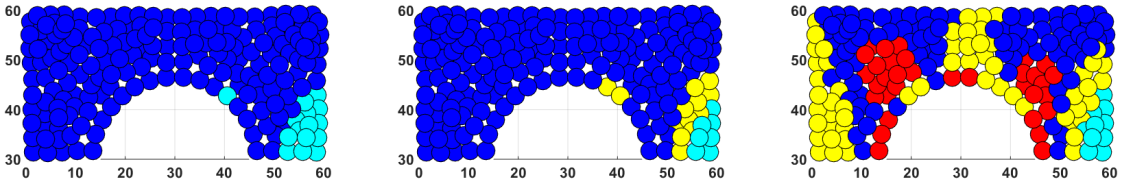
The factors obtained from applying PCA on the stress and the transformed safety factor data are clustered using K-means and Ward's method. We use both clustering techniques for a brief comparison. We also cluster factors based on both stresses and safety factor data to have an insight at the end about which mechanical measure would lead to a better domain partitioning and accordingly better optimization results.

In the following, each factor is divided into a different number of sub-clusters. In Fig. 5.10 and Fig. 5.11, we show two, three, and four divisions for each factor of the stress raw data previously shown in Fig. 5.8 using K-means and Ward's method respectively. Similarly, in Fig. 5.12 and Fig. 5.13, we show two, three, and four divisions for each factor of the transformed safety factor raw data in Fig. 5.9 using K-means and Ward's method respectively. Note that each finite element belongs to one of these sub-clusters and is marked with a circle at the centre of the element. We chose a maximum number of divisions equal to four for each factor to reduce the computational burden. After clustering the factors, the idea is to intersect them together for every possible combination of sub-divisions so we could get a set of global sub-divisions of the domain. We will call each intersection of the four factors a partition and we will denote it by $\mathcal{P}$ where all the partitions belong to a set $\mathbb{P}$. The size of the set $\mathbb{P}$ varies with the number of divisions per factor and the total number of factors chosen. For example, if four factors are sub-clustered up to eight divisions, the set $\mathbb{P}$ would possess 8^4 partitions (possible combinations).

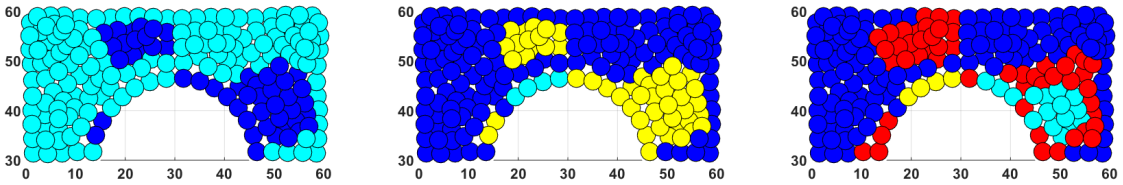
From Figs. 5.10 to 5.13, we show the division of the factors based on the stress and the safety factor raw data using K-means and Ward's clustering techniques. We could detect just by eye-balling the figures that both clustering techniques behave almost similarly. We could also see that it is possible to have some elements belonging to the same cluster but disconnected in the spatial domain. This is because we cluster the data based on the values of the factors obtained from PCA not based on the location of the element in the domain.



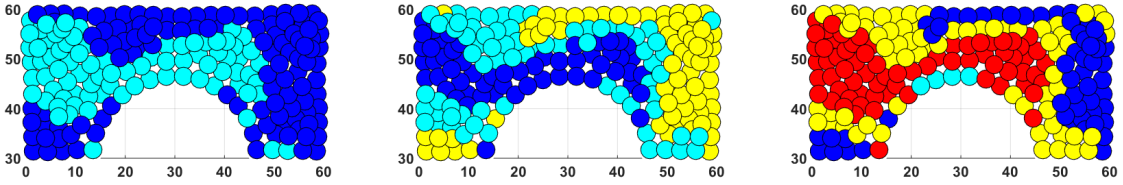
(a) Factor 1



(b) Factor 2

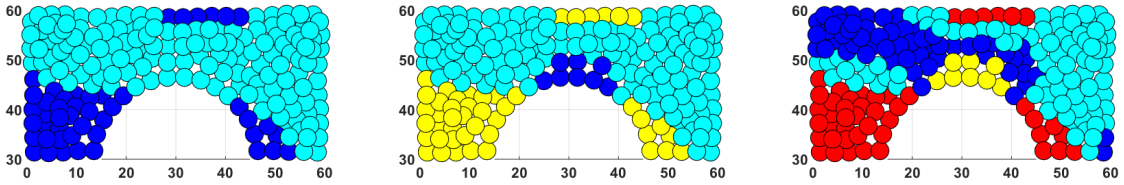


(c) Factor 3

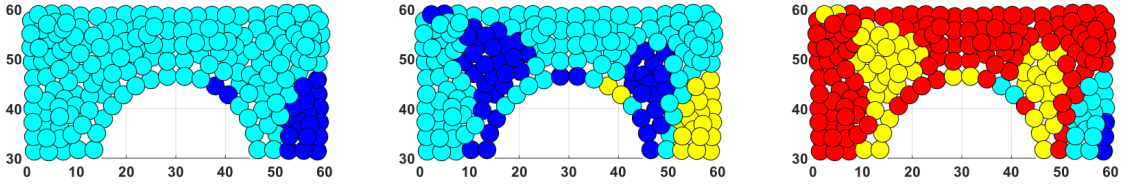


(d) Factor 4

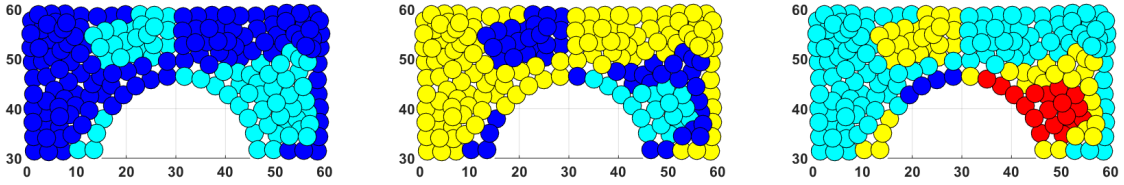
Figure 5.10: Clustering of the first four factors obtained from applying PCA on the stresses raw data matrix using K-means. Each factor is divided into 2,3,4 clusters from left to right.



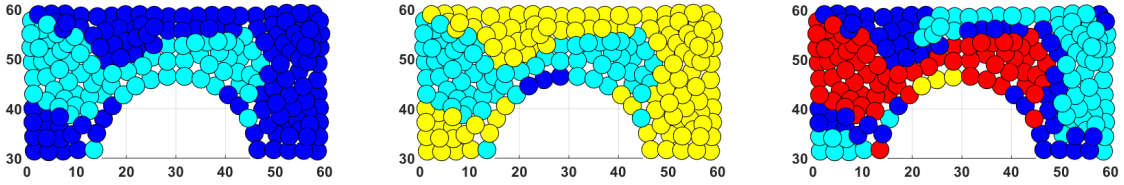
(a) Factor 1



(b) Factor 2

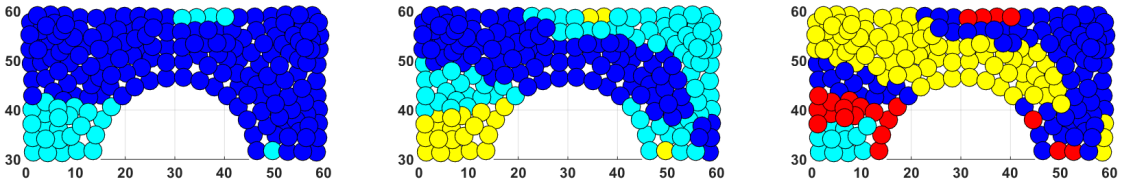


(c) Factor 3

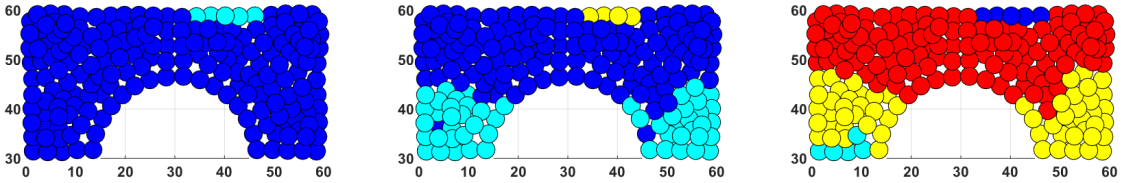


(d) Factor 4

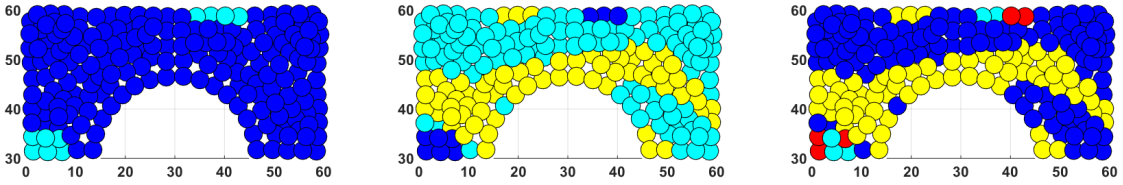
Figure 5.11: Clustering of the first four factors obtained from applying PCA on the stresses raw data matrix using Ward's method. Each factor is divided into 2,3,4 clusters from left to right.



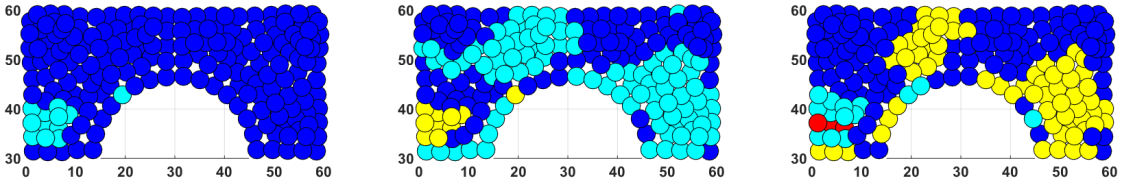
(a) Factor 1



(b) Factor 2

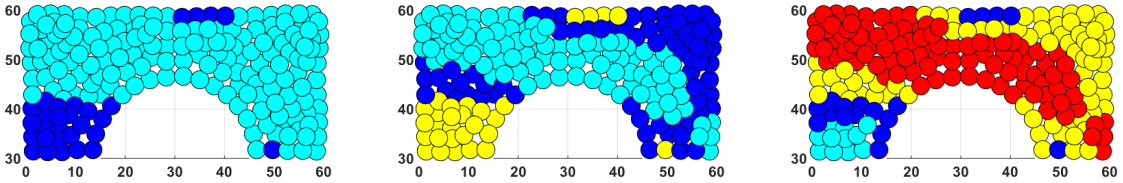


(c) Factor 3

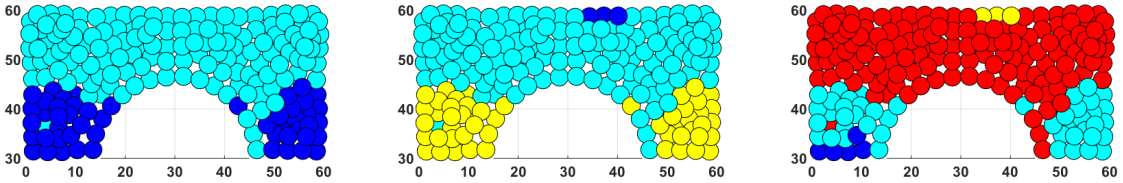


(d) Factor 4

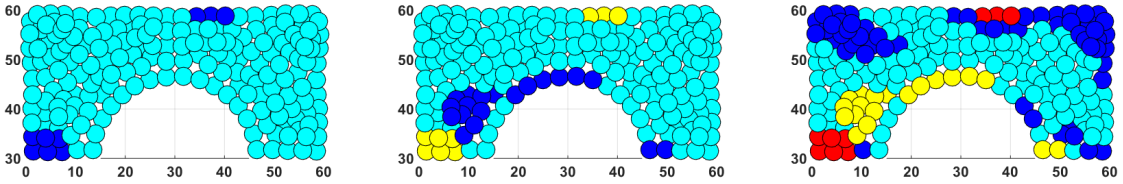
Figure 5.12: Clustering of the first four factors obtained from applying PCA on the transformed safety factor raw data matrix using K-means. Each factor is divided into 2,3,4 clusters from left to right.



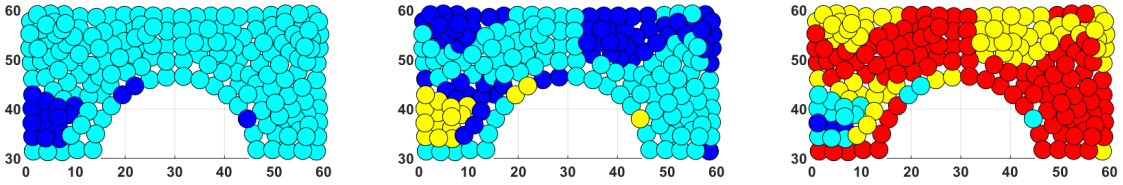
(a) Factor 1



(b) Factor 2



(c) Factor 3



(d) Factor 4

Figure 5.13: Clustering of the first four factors obtained from applying PCA on the transformed safety factor raw data matrix using Ward's method. Each factor is divided into 2,3,4 clusters from left to right.

At this point we have clustered the first four factors obtained from PCA based on the stress data and the transformed safety factor data. The next step is to intersect those factors together to get all the possible combinations of partitions $\mathcal{P}$ that will constitute the set of partitions $\mathbb{P}$. Since we adopt four factors, based on Fig. 5.7, as it was enough to represent the data and each factor was sub-divided maximum into four clusters; then the possible partitioning from intersecting the factors ranges from 1 to 4^4 possibilities.

An example of intersection is shown Fig. 5.14, where the number of clusters for factors 1,2,3,4 is 1,2,1,4 respectively. Note that we neglected factors one and three in Fig. 5.14 since they only have 1 cluster each.

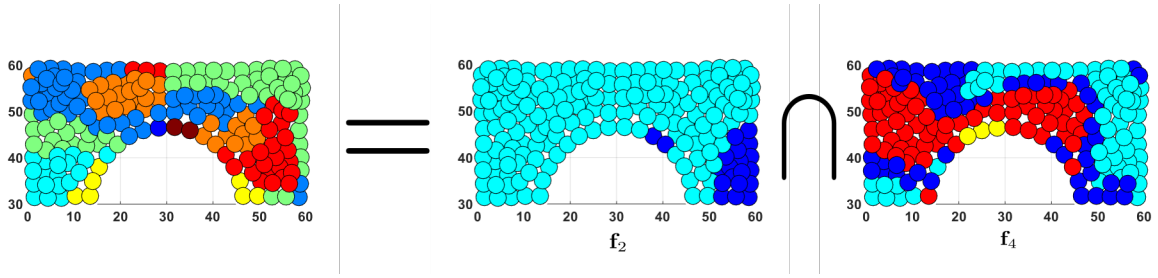


Figure 5.14: Partition with 8 clusters resulting from the intersection of 4 factors where each of them was clustered using Ward's method. The number of clusters for factors 1,2,3,4 is 1,2,1,4 respectively.

5.2.4 Error computation and clustering optimization

In order to find the "best" intersection of factors we need to perform a clustering optimization based on a given error measure. The idea is to find the partition possessing the minimum within cluster variance, leading to more homogeneous partitions in terms of the mechanical measure used. In order to achieve such goal, we use the Sum of Squares Error (SSE). SSE is a measure of discrepancy between the data of an element (the stress data or the safety factor data) and the average of the data in the cluster where the element belongs. Thus, for a domain having a number of clusters as many as the number of elements, the SSE measure is equal to zero. Moreover, if all the elements in a cluster are the same, then also the SSE would be zero. SSE will be used as an optimality criterion in our problem such that the clusters in a given partition possess elements that are very similar to each other. The smaller is the SSE measure, the better is the partitioning of the domain and the clusters are more uniform.

Following the work done by Alaimo *et al.* [22], we will show how the SSE is computed for the stress raw data knowing that the same procedure is followed for any kind of raw data matrix. We first define the total maximum error over all the stress components of the data matrix $\tilde{\boldsymbol{\sigma}}$ yielding,

$$E_{max} = \sum_{s=1}^{\mathtt{N_c}} \sum_{i=1}^{\mathtt{n_{el}}} (\tilde{\sigma}_i^s - \overline{\tilde{\sigma}^s})^2, \tag{5.4}$$

where the $\overline{\tilde{\sigma}^s}$ is the mean of the s-th stress data matrix component. We compute now the error in each sub-cluster of a given partition $\mathcal{P}$ having more than one cluster in it.

The error associated with the s-th component of $\tilde{\sigma}^s$ reads,

$$E^s(\mathcal{P}) = \frac{1}{E_{max}} \sum_{\ell=1}^{\mathtt{n_s}} \sum_{i=1}^{n_\ell(\mathcal{P})} (\tilde{\sigma}_i^s - \overline{\tilde{\sigma}^s_{,\ell}})^2, \tag{5.5}$$

where $n_\ell(\mathcal{P})$ and $\overline{\tilde{\sigma}^s_{,\ell}}$ are the number of elements and the mean value of the s-th component of the stress within the ℓ-th cluster, respectively; and $\mathtt{n_s}$ is the total number of clusters in each partition $\mathcal{P}$. Now, we could compute the global error as follows:

$$E(\mathcal{P}) = \sum_{s=1}^{\mathtt{N_c}} E^s(\mathcal{P}) \quad = \frac{1}{E_{max}} \sum_{s=1}^{\mathtt{N_c}} \sum_{\ell=1}^{\mathtt{n_s}} \sum_{i=1}^{n_\ell(\mathcal{P})} (\tilde{\sigma}_i^s - \overline{\tilde{\sigma}^s_{,\ell}})^2. \tag{5.6}$$

Exactly a similar procedure is followed to evaluate the SSE for the partitions obtained from the intersection of the factors that are based on the transformed safety factor raw data. It is worth noting that if each finite element is a cluster on its own ($\mathtt{n_s} = \mathtt{n_{el}}$), we get the lowest global SSE ($E(\mathcal{P}) = 0\%$) while the maximum global SSE ($E(\mathcal{P}) = 100\%$) is when partition $\mathcal{P}$ consists of only one cluster. Consequently, it is a trade off between the global SSE and the number of clusters because the more we decrease the SSE, the number of cluster increases and vice versa. Accordingly, the goal is to simultaneously minimize the total global SSE, $E(\mathcal{P})$, to have accurate results and minimize the number of clusters for computational efficiency. According to Alaimo *et al.* [22], the clustering multi-objective optimization problem could be defined as follows:

Let the superscript "$\mathtt{Opt}$" indicate optimality, find $\mathcal{P}^{\mathtt{Opt}}$ such that,

$$\mathcal{P}^{\mathtt{Opt}} = \arg\min_{\mathcal{P}} \ \{E(\mathcal{P}), \mathtt{n_s}(\mathcal{P})\} \quad \text{s.t.} \quad \mathcal{P} \in \mathbb{P}. \tag{5.7}$$

As argued in [22], there is not a single partition that represents a global minimum of the problem which leads to a *Pareto optimality* situation. A Pareto optimality is a situation where improving a criterion cannot happen without making another criterion worse. What we are aiming for is to obtain a *Pareto set* or *Pareto front* which is the set having optimization solutions that are superior to the rest of the solutions in the search space $\mathbb{P}$ while the solutions among the Pareto set do not dominate each other [90]. Note that, a partition $\mathcal{P}_1$ is said to dominate another partition $\mathcal{P}_2$ only when the following inequalities hold [22]:

$$E(\mathcal{P}_1) \leq E(\mathcal{P}_2) \quad \text{and} \quad \mathtt{n_s}(\mathcal{P}_1) \leq \mathtt{n_s}(\mathcal{P}_2) \tag{5.8a}$$

$$E(\mathcal{P}_1) < E(\mathcal{P}_2) \quad \text{or} \quad \mathtt{n_s}(\mathcal{P}_1) < \mathtt{n_s}(\mathcal{P}_2). \tag{5.8b}$$

The optimization problem in (5.7) is solved with Algorithm 5.1.

Algorithm 5.1: Pseudo-code for clustering optimization algorithm

Input : Set of partitions $\mathbb{P}$, Maximum number of clusters, stress raw data, transformed safety factor raw data

Output: Clustering solutions, Pareto set

(1) Set $\mathtt{n_s} = 1$

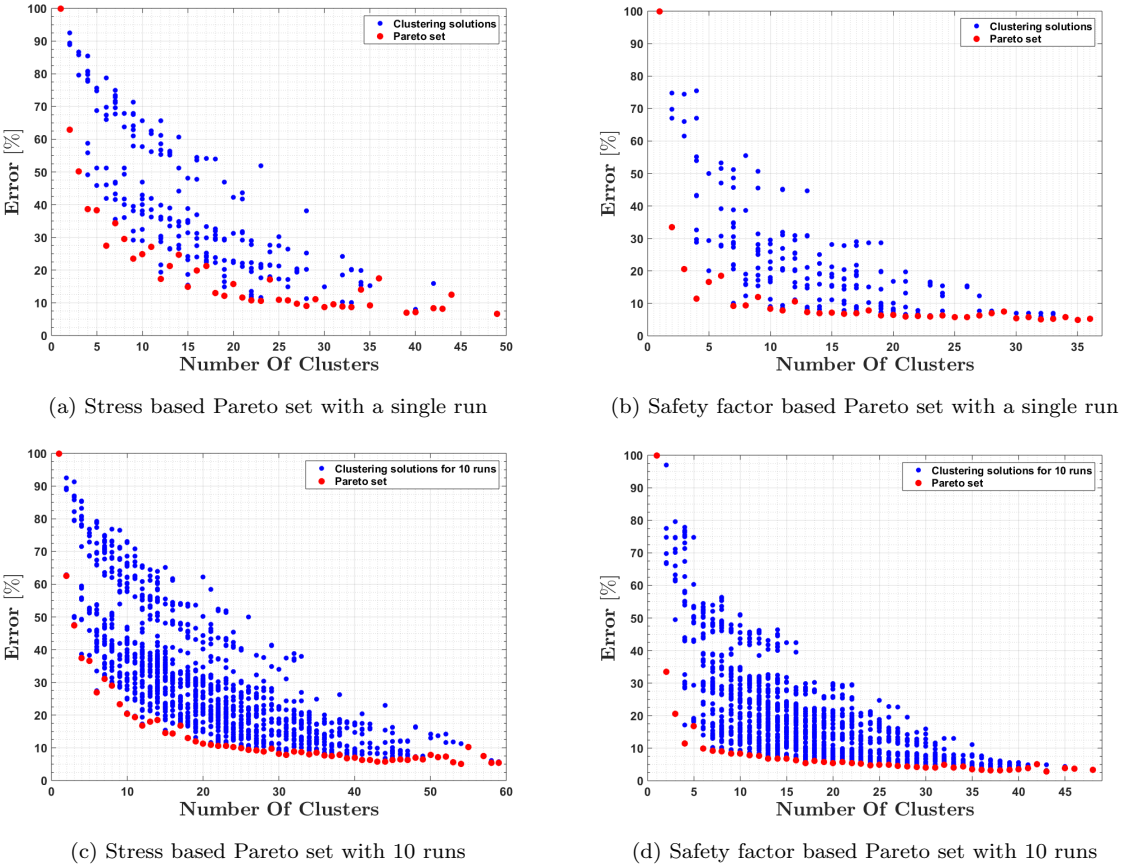
(2) **while** $n_s \leq$ *Maximum number of clusters* **do**

- (2.1) Extract all the partitions $\mathcal{P}$ from the set $\mathbb{P}$ that are sub-divided into $\mathtt{n_s}$ clusters
- (2.2) Compute the corresponding global error $E(\mathcal{P})$ as shown in (5.6)
- (2.3) Store the errors of all the partitions

(3) Apply Eq. (5.8a) and Eq. (5.8b) to obtain the Pareto set

We run the algorithm for both the K-means and Ward's method clusters. The output from the algorithm is all the possible clustering solutions based on the set of partitions $\mathbb{P}$, as shown in Fig. 5.15 and Fig. 5.16, where the blue dots represent the clustering solutions and the red ones represent the the Pareto set. As explained in Appendix D.1, K-means algorithm is very efficient, however, it has got an element of randomness due to the random initialization of the centroids. A common practice to overcome this issue and to ensure we converge to the correct results is to run the K-means algorithm several times and take the best output. This leads only to a bigger set of partitions $\mathbb{P}$, but the procedure is exactly the same.

In our problem, we have run the K-means 10 times and monitored the clustering solutions as shown in Fig. 5.15c and Fig. 5.15d. It was found that there is a very slight improvement in terms of the global error $E(\mathcal{P})$ as shown in Table 5.1.



(a) Stress based Pareto set with a single run

(b) Safety factor based Pareto set with a single run

(c) Stress based Pareto set with 10 runs

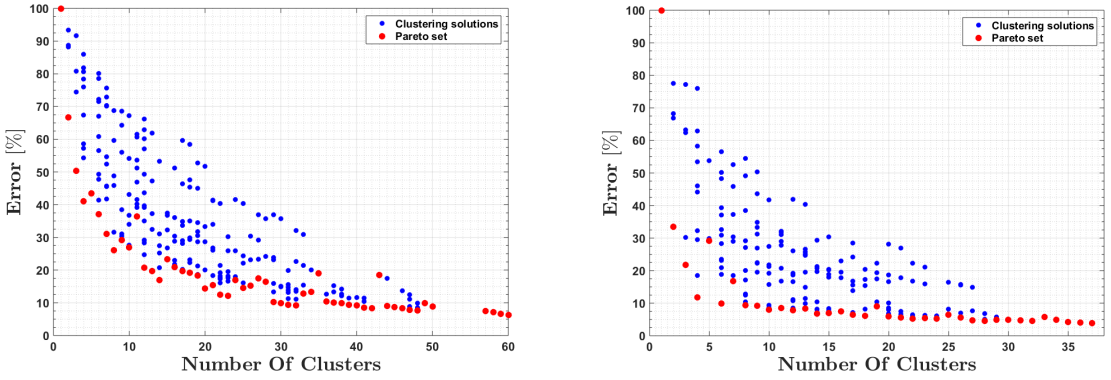
(d) Safety factor based Pareto set with 10 runs

Figure 5.15: Clustering optimization solutions obtained from Algorithm 5.1 using K-means with a single run and 10 runs.

,

Pareto set error comparison				
	Stress based clusters		Safety factor based clusters	
	4 clusters	8 clusters	4 clusters	8 clusters
K-means single run	38%	29%	11.5%	9.5%
K-means 10 runs	37%	29%	11.2%	9.1%
Ward's method	41%	26%	11.9%	9.3%

Table 5.1: Pareto set error comparison between K-means with a single run, K-means with 10 runs, and Ward's method



(a) Stress based Pareto set

(b) Safety factor based Pareto set

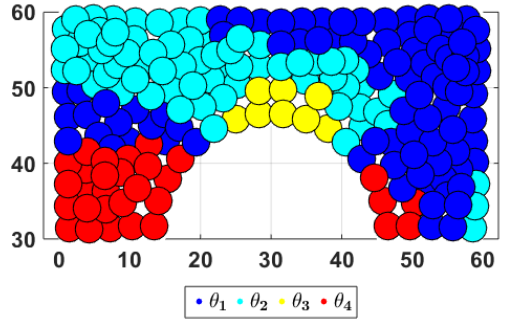
Figure 5.16: Clustering optimization solutions obtained from Algorithm 5.1 using Ward's method

It can be deduced by visually comparing Fig. 5.16 with Fig. 5.15 and from the errors shown in Table 5.1 that Ward's method's results are very close to the ones obtained by K-means. Consequently, it is essential to test the clustering solutions obtained from both methods on the original PGD problem to decide which clustering technique is performing better for our optimization problem. The idea now is to run the PGD with the parameterized fibre orientations assigned to partitions obtained from the clustering results. The outcome of the test would be the optimal fibre orientation and we will also compare it with the optimal fibre orientation obtained from the PGD without the clustering analysis presented in Chapter 4.

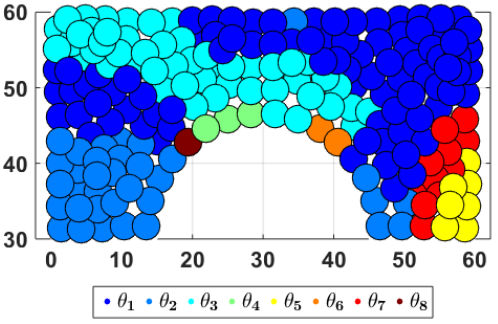
5.2.5 Fibre orientation optimal results

In the current section, we show the optimal fibre orientation solutions obtained using PGD, where the domain is partitioned based on the obtained optimized clusters. In Figs. 5.17 and 5.18, we illustrate the domains with four and eight parameters (subdomains) that were in the Pareto set obtained based on the K-means and Ward's method clustering optimization of both the stress and the transformed safety factor data. The clustered domains are used as input for the PGD, and the different clusters are parameterized with different independent fibre orientations to be optimized.

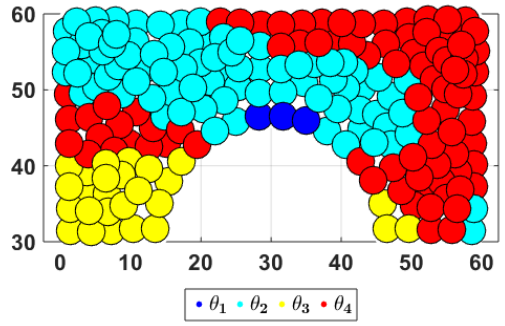
It was found out that the optimal fibre orientations results, obtained using PGD, based on the domains in Figs. 5.17a, 5.17d, 5.18a and 5.18b are better than the ones obtained in Chapter 4 in the examples with the domains partitioned based on intuition (as in Fig. 4.18 and Fig. 4.21). This outcome results from the measurement of the failure onset index, that is the safety factor index (λ_s) explained in Chapter 2 by Eq. (2.42), corresponding to the optimal fibre orientation. The comparison between different findings based on different partitioning of the domain is shown in Table 5.2. Recalling that the optimization objective in Chapter 2 consists in maximizing the safety factor index λ_s,



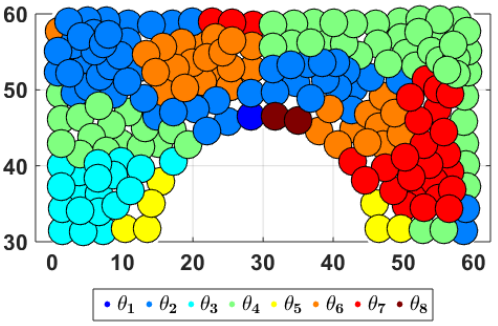
(a) K-means with 4 parameters



(b) K-means with 8 parameters



(c) Ward's with 4 parameters



(d) Ward's with 8 parameters

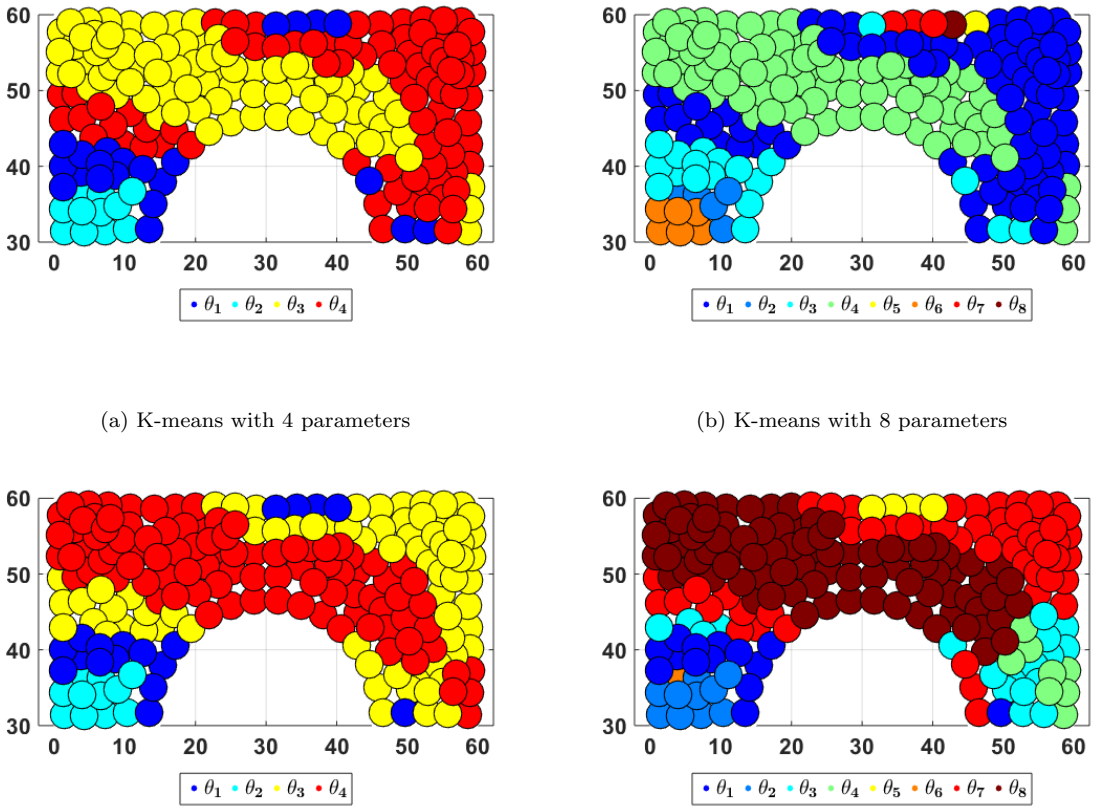
Figure 5.17: Domain parameterization for PGD solver based on the clustering optimization of the stress data using: K-means in (a) & (b) and Ward's method in (c) & (d)

therefore, the higher the value of λ_s the better our results are.

	Domain with 4 parameters	Domain with 8 parameters
Stress based clustering with K-means	0.7863	0.8788
Stress based clustering with Ward's	0.8653	0.9037
Transformed safety factor clustering with K-means	**1.013**	**0.9934**
Transformed safety factor clustering with Ward's	0.7973	0.8244
Based on intuition (Chapter 4)	0.8254	0.879

Table 5.2: Safety factor index λ_s obtained from PGD based on different domain parameterization shown in Fig. 5.17 and Fig. 5.18

As shown in Table 5.2, the PGD result based on the clustered domains show a significant improvement. There is an increase of 23% in the safety factor index value, for the 4 parameter domain, whereas there is an increase of 13% in the safety factor index value,



(a) K-means with 4 parameters

(b) K-means with 8 parameters

(c) Ward's with 4 parameters

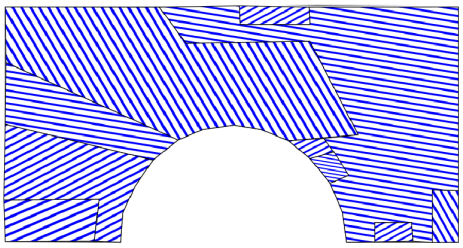
(d) Ward's with 8 parameters

Figure 5.18: Domain parameterization for PGD solver based on the clustering optimization of the transformed safety factor data using: K-means in (a) & (b) and Ward's method in (c) & (d)

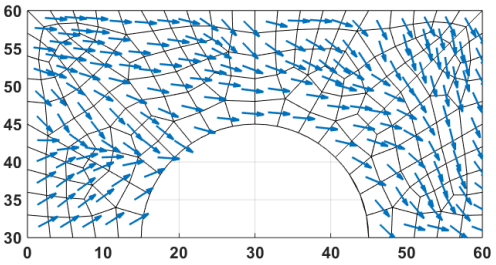
for the 8 parameter domain, compared to the results obtained in Chapter 4. The optimal fibre orientation resulting from PGD based on the domains in Figs. 5.18a and 5.18b is shown in Figs. 5.19a and 5.19c. It can be seen that there are jumps in the fibre orientation between patches. In an attempt to reduce fibre discontinuities and avoid big angle jumps between elements with the aim of having a more homogeneous distribution of fibres, we resort to the application of a filter as follows:

$$\tilde{\boldsymbol{\theta}} = \boldsymbol{H}\boldsymbol{\theta}, \tag{5.9}$$

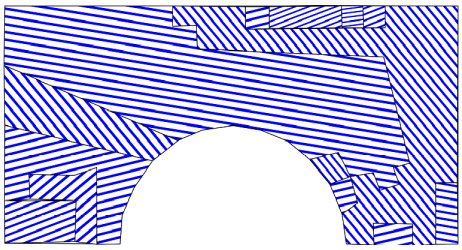
where $\tilde{\boldsymbol{\theta}}$ is the $\mathtt{n_{el}} \times 1$ filtered angle vector and $\boldsymbol{H}$ is the $\mathtt{n_{el}} \times \mathtt{n_{el}}$ filter matrix. The filter matrix holds the weight coefficients relating the filtered angle at a given element



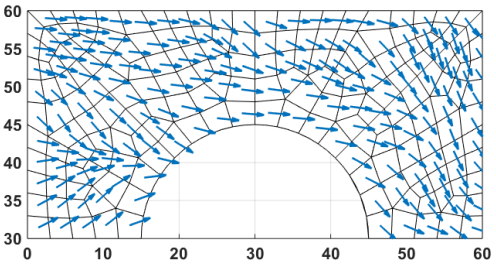
(a) Optimal discontinuous fibres for domain in Fig. 5.18a



(b) Filtered fibre orientation corresponding to (a)



(c) Optimal discontinuous fibres for domain in Fig. 5.18b



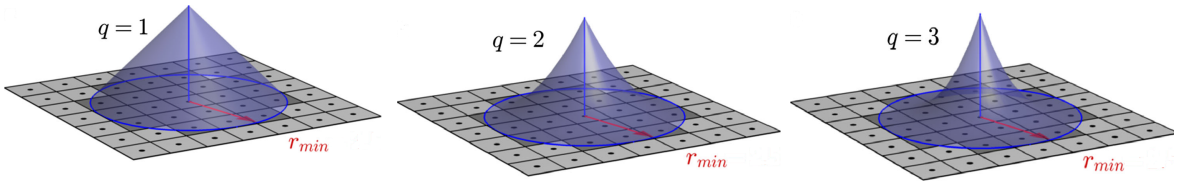
(d) Filtered fibre orientation corresponding to (c)

Figure 5.19: Example of optimal fibre orientation for domains with 4 and 8 parameters obtained using the clustering of the safety factor via K-means

with the angles at the neighbouring elements and it has the following form:

$$\begin{aligned} \boldsymbol{H}_{ij} &= \frac{h(i,j)\mathsf{v}_j}{\sum_k^{\mathtt{n_{el}}} h(i,k)\mathsf{v}_k} \\ \text{with} \qquad h(i,j) &= \max\left\{0, [r_{min} - \mathsf{dist}(i,j)^q]\right\}, \end{aligned} \tag{5.10}$$

where v_i is the volume of the i-th element, r_{min} is a user defined radius defining the application region of the filter, $\mathsf{dist}(i,j)$ is a distance measure between $\tilde{\theta}_i$ and θ_j, and finally q is an exponent defining the order of the filter's weighting function as shown in Fig. 5.20. The output for a radius of $r_{min} = 10$, which is almost including three neighbouring elements, is shown in the schematics in Fig. 5.19b and Fig. 5.19d where the arrows originating from the centre of each element indicate the direction of the fibres at the corresponding element.



(a) Linear filter (b) Quadratic filter (c) Cubic filter

Figure 5.20: Weighting functions example. Illustration adapted from [3]

5.3 Introduction to Additive Manufacturing (AM) and experimental validation

In the previous sections it was shown the importance of clustering analysis, and it was shown that it has a direct impact on the optimal fibre orientation in composite laminates obtained using PGD. It is also shown that the new optimal solutions are highly complex in terms of fibre orientation with large discontinuities between sub-domains which makes its manufacturing challenging. Thanks to the Additive Manufacturing (AM) technology (also called 3D printing technology), this kind of complex domains could be realized in a short amount of time with very high precision.

The AM process is illustrated in Fig. 5.21 (illustration adapted from [91]) and it consists of five main steps. It starts with a 3D CAD model where the part to be printed is designed. The CAD file is then converted to an stereolithography file (STL file). The STL file describes the triangulated surface of the 3D object in terms of unit normals and vertices. The STL file is then introduced to another software called a slicer. The slicer's job is to take the 3D object and slice it into 2D layers. The slicing step is of paramount importance as it also defines the path by which the layers will be printed. The slicing process outputs a file called the G-Code which holds the set of instructions that needs to be sent to the 3D printer in order to manufacture the part. The G-Code file contains valuable information obtained from the slicing process such as the printing path, printing patterns, layer thickness, and printing density, to name a few. Once the G-Code is passed to the 3D printer, the machine starts to build the actual 3D model layer by layer until the part is completed. A final step is to finish and post-process the part and the type of post-processing depends on the printing technology used.

There are many types of AM technologies nowadays in the industry, and they could be classified based on the way the material is fed to the printer, the type of materials used, or layer formation technique. Examples of AM technologies are Stereolithography (SLA),

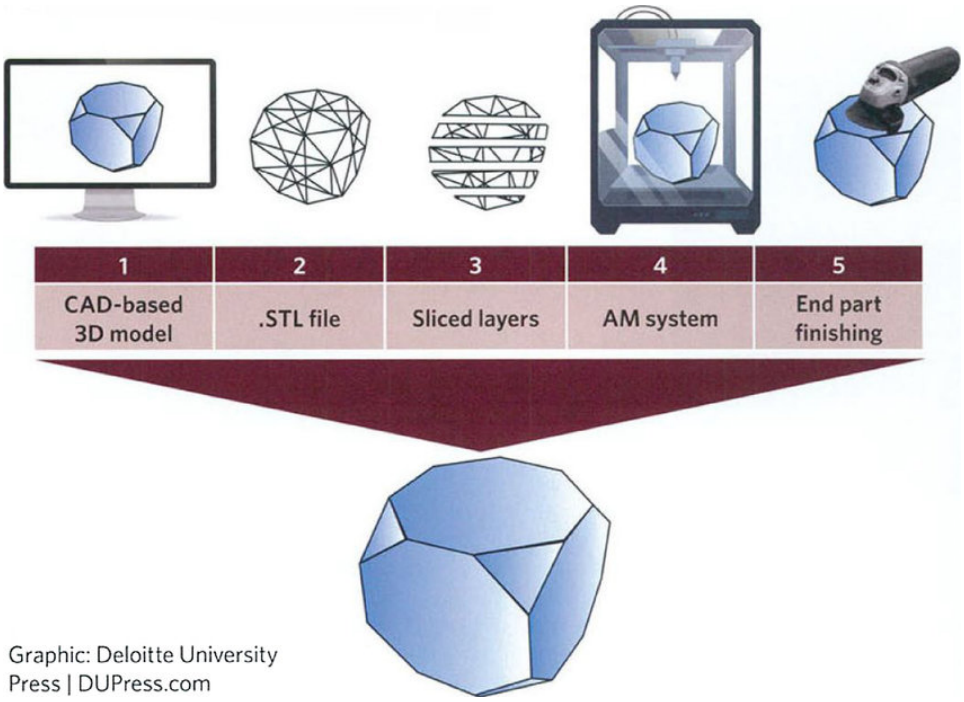


Figure 5.21: The Additive Manufacturing (AM) process.

Fused Deposition Modelling (FDM), Selective Laster Sintering (SLS), Laminated Object Manufacturing (LOM), to name a few. The interested reader should refer to [92] and the references therein. Since our work in this thesis is on fibrous composite laminates, then we will briefly introduce the FDM technology which uses filament deposition to produce the printed components and it is one of the most widely used technologies for 3D printing [93].

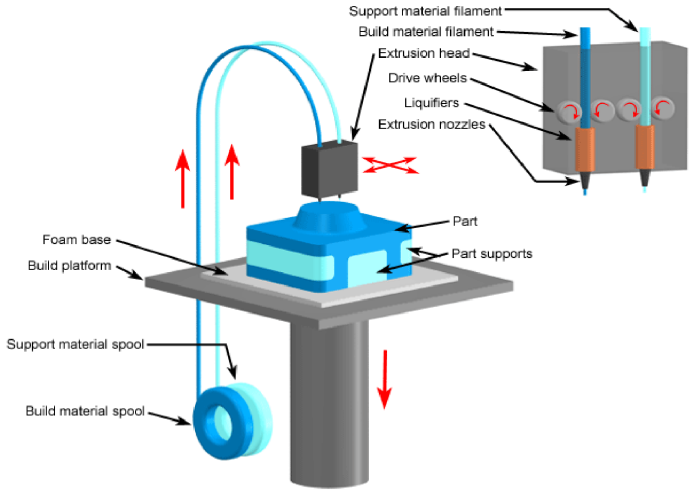


Figure 5.22: FDM technology system scheme

FDM is by far the most common AM technology that is based on extrusion of material [92]. In Fig. 5.22, we can observe the complete system scheme. FDM uses thermoplastic molten filament to feed the system and the filament is pushed via a tractor-pulley system which generates the extrusion pressure. Most FDM based machines have two sets of extruder heads with, for example, one for a thick nozzle and another for a thinner one. The filament is then deposited on the bed of the machine where it rests there to solidify. The deposition of the material in one layer occurs in two main steps. First, a matrix defining the borders of the layer is deposited, and then the material filling of that matrix also referred to as *infill*. In the same manner, the material is deposited for each layer until the desired part is ready. An important design consideration when using FDM is to take into account the mechanical properties of the part that change due to the filament deposition orientation. By nature, the filaments are stronger in one direction over the other and, therefore, the filling pattern is crucial to the design.

The definition of the filament infill patterns is part of the slicing process. There exists many slicing commercial software with built-in predefined printing patterns, such as rectilinear, concentric, triangular, honeycomb, grid, and zigzag as shown in Fig. 5.23.

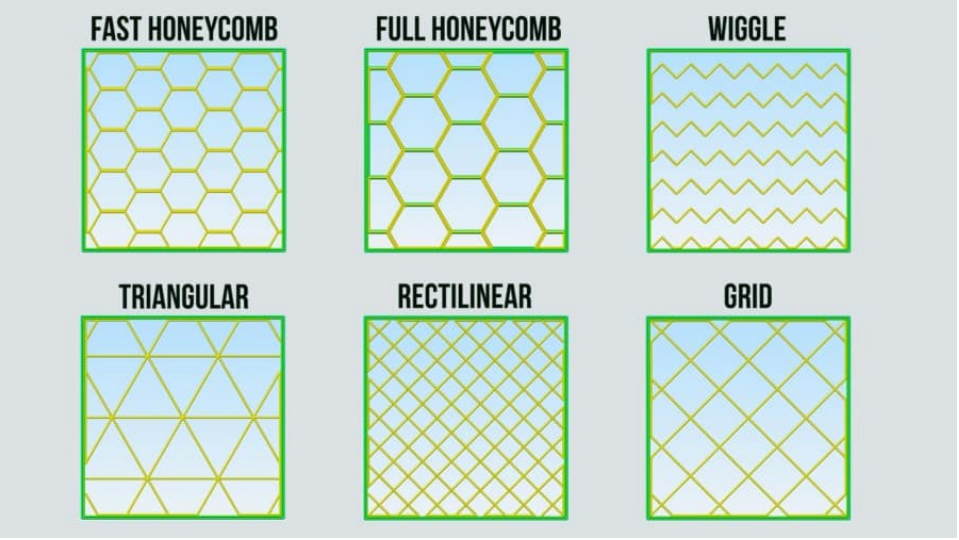
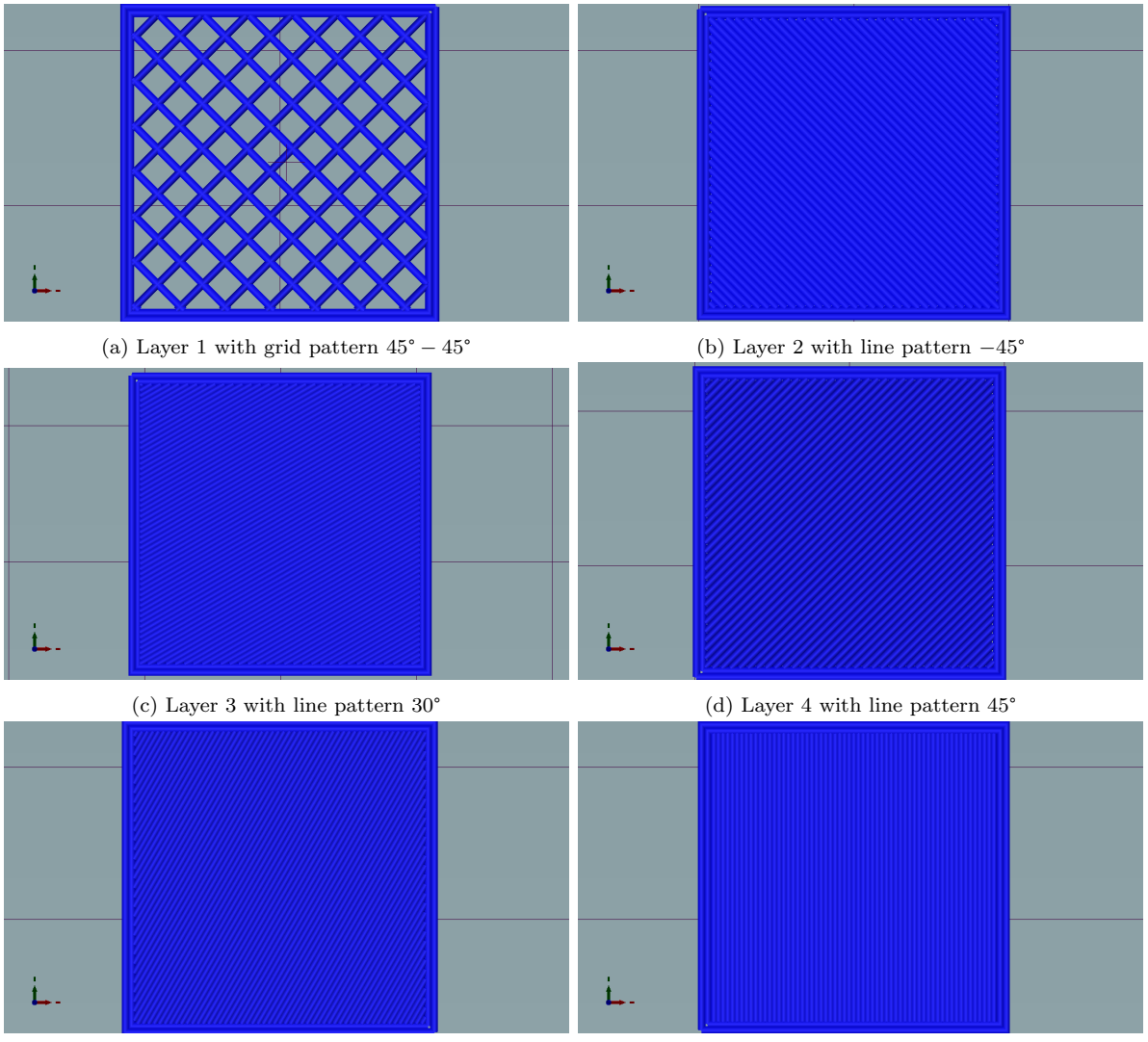


Figure 5.23: Common infill patterns found in commercial software

The predefined patterns give the designer a limited choice when taking into account the mechanical properties of the component. As it was shown in the previous sections that the optimal fibre orientation could have a complex shape and discontinuous regions of fibres with different orientation. In an attempt to explore the possibility of obtaining customized infill patterns based on the optimization undergone in the past sections, we took a preliminary step to modify the open source C++ slicing software CuraEngine. The modification is very simple, it consists in modifying the infill function in the code so we could get a different infill pattern for each layer based on a given list of angles. We tried the modified function on a cube with six layers, where we have one layer of a grid type

pattern of $-45°/45°$ and the rest of the stacking sequence is $-45°/30°/60°/45°/90°$ for layers 2,3, 4, 5, and 6 respectively. The output is generated using Repetier-host© for illustration and it is shown in Fig. 5.24.



(a) Layer 1 with grid pattern $45° - 45°$

(b) Layer 2 with line pattern $-45°$

(c) Layer 3 with line pattern $30°$

(d) Layer 4 with line pattern $45°$

(e) Layer 5 with line pattern $60°$

(f) Layer 6 with line pattern $90°$

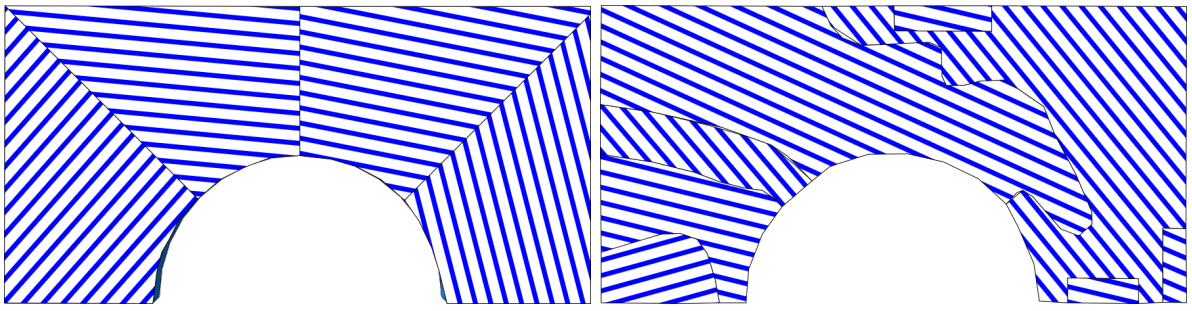
Figure 5.24: Example of a cube with custom infill pattern generated by modifying the slicing software CuraEngine

By doing so, we open the door for editing the slicing software to get a specific desired pattern and possibly continuous fibre orientation which is a very active area of research nowadays [94], however, this will not be in the scope of this thesis and is considered as part of the future work.

In the following we present the output of preliminary experimental tests undergone for the specimen simulated in the present chapter. The experimental tests are crucial for the validation of the clustering optimization and the methodology in general. At this point, we have already obtained the optimal results as shown in Figs. 4.20 and 5.19a.

The aim of the tests is to experimentally demonstrate that the optimal results obtained using a domain partitioned based on clustering are better than the ones obtained using a symmetrically partitioned domain. The comparison between the domains occurs by monitoring the loads at which the specimens exhibit failure during the tensile test. The experimental validation is done in three main steps:

(1) **Simulation and analysis:** The first step is to run the model, which we have already done by now, to obtain the optimal fibre orientation in different domains (shown in Fig. 5.25).
(2) **Specimen preparation and 3D printing:** The specimen preparation consists in designing the grips of the component where the tensile testing machine would clamp the component (shown in Fig. 5.26). It also consists in preparing the STL files of the components to be printed and slicing the part for the G-Code generation.
(3) **Tensile test and monitor results:** Set up the 3D printed part on the tensile testing machine, perform traction until first point failure occurs, and then record the corresponding load for comparison.



(a) Symmetrical partitioning (b) Clustering based partitioning

Figure 5.25: Optimal fibre orientation for domains based on symmetrical partitioning and on clustering results using ABS material

So far in the current thesis, we have been using carbon fibre reinforced ABS material data throughout all our simulations. However, for the sake of 3D printing and testing, and due to material availability, the material used in the test is ABS instead of CF-ABS and , therefore, we simulated the same problem for different material properties corresponding to ABS characteristics that are shown in [95]. The optimal results obtained from PGD for a symmetrically partitioned and cluster based partitioned domains using ABS material properties are shown in Fig. 5.25.

The specimen preparation step consists in designing extra parts at the extremities of the structure where the testing machine would grip as shown in Fig. 5.26. The grip section in the specimen should be tapered not to have any stress concentrations. Furthermore,

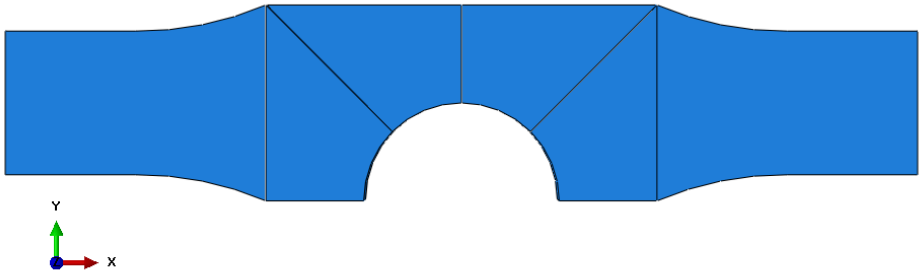
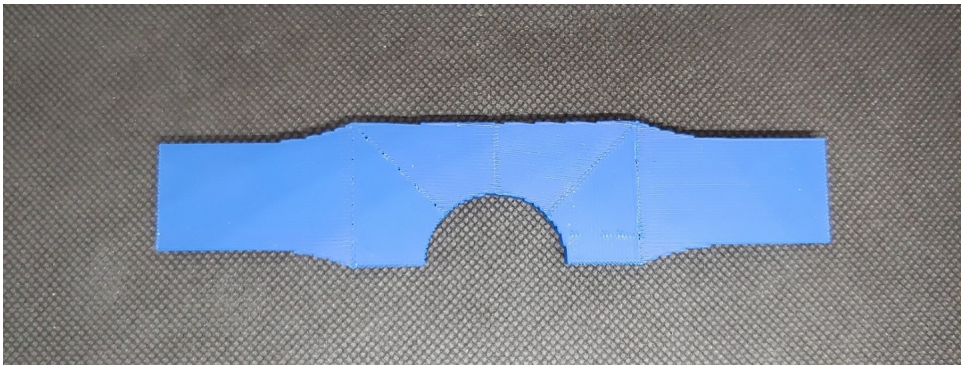
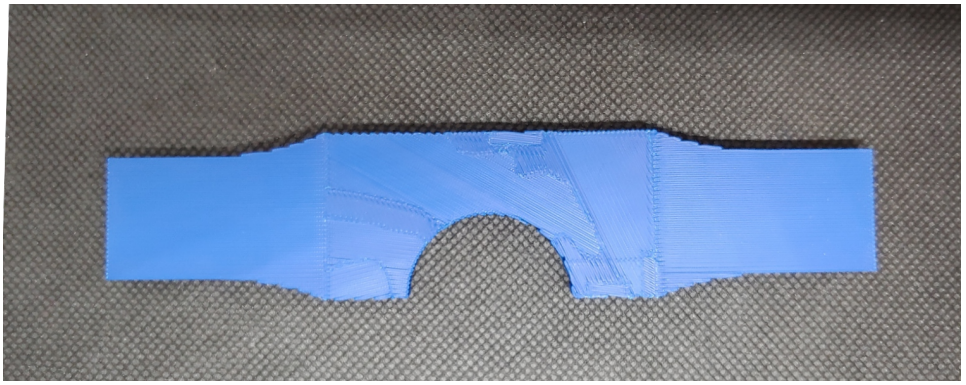


Figure 5.26: Component designed with grips for the tensile test

there is the need to provide the slicing software with different STL files where each file corresponds to a different zone in the structure having a different fibre orientation. We used the commercial software Abaqus to perform the aforementioned preparation steps.



(a) Symmetrical partitioning



(b) Clustering based partitioning

Figure 5.27: Optimal fibre orientation for domains based on symmetrical partitioning and on clustering results using ABS material

The slicing of the parts is performed using CuraEngine slicing software. We provide the slicing software with the STL files and their corresponding fibre orientations, and we

ultimately obtain the G-Codes. In Fig. 5.27, we show the 3D printed components for both the manually partitioned and the cluster based partitioned parts.

The tensile test undergone consisted in pulling the specimens from both grips until failure. The test results show great agreement with the methodology showing an enhancement in the load bearing capacity of the structure. The load against extension curve is shown in Fig. 5.28. We could observe that the structure optimized based on the clustering analysis partitioning breaks at a load $\sim$ 1370 N where the structure that was partitioned manually breaks at a load $\sim$ 1070 N yielding a 22% increase in the load carrying capacity. The failure in each of the structures occurs suddenly in a brittle-like failure mode due to the discontinuity of fibres between zones as shown in Fig. 5.29. This confirms the compelling need for continuous fibre manufacturing to avoid breaks at the interfaces between zones in specimens.

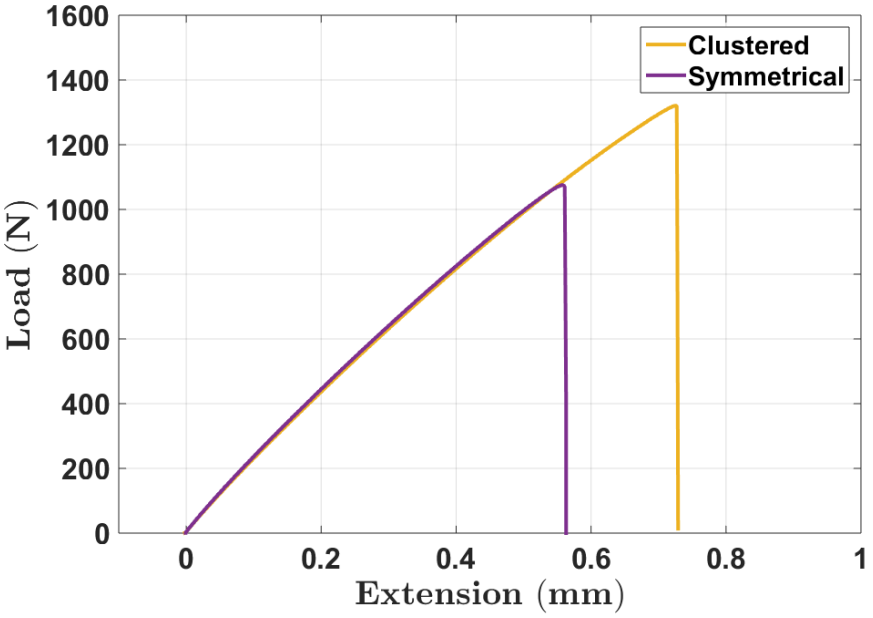
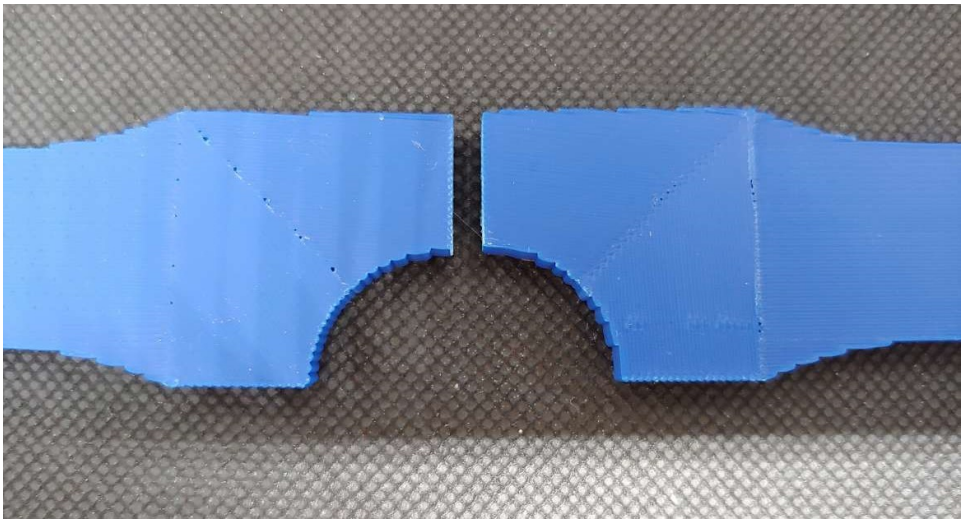
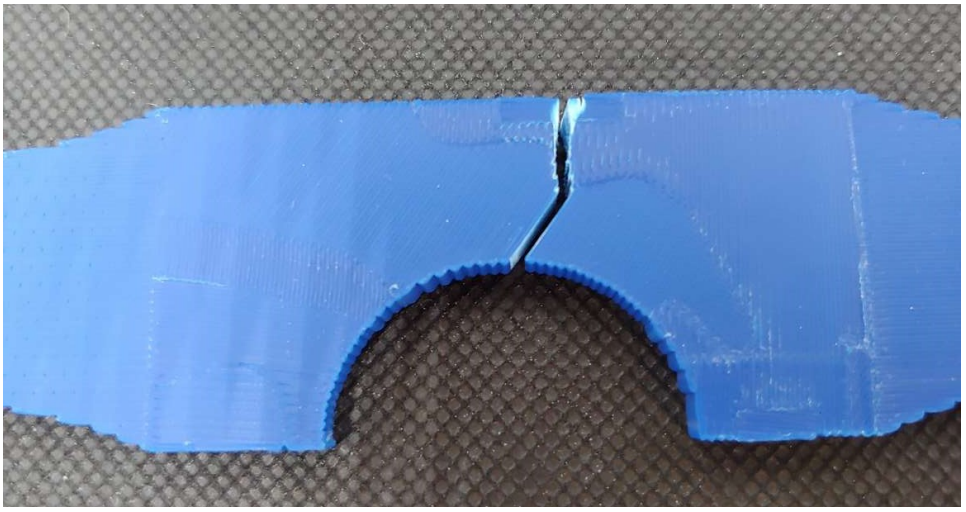


Figure 5.28: Load vs extension curves for specimens with symmetrical and clustered based partitioning



(a) Symmetrical partitioning



(b) Clustering based partitioning

Figure 5.29: Failure of structure for domains based on symmetrical partitioning and on clustering resulting from tensile tests

Chapter 6

CONCLUDING REMARKS

6.1 Summary

The current dissertation presents a new methodology with the intention of obtaining generalized solutions for the deformation of composite laminates parameterized with fibre orientations using the Proper Generalized Decomposition (PGD) framework. The aim is to optimize the fibre orientation in composite laminates, with an affordable computational cost, to improve the macro-mechanics of the laminate. This is relevant for the mechanical optimization of additively manufactured components. The main ingredients of this multidisciplinary research include solid mechanics, failure theories, numerical methods, optimization techniques, reduced order modelling techniques, additive manufacturing, experimental testing, coupling and development of computer codes, and machine learning techniques for data analysis. In the following, we present some concluding remarks on the work presented and a list of future work ideas.

6.2 Concluding remarks

The problem statement including the governing equations, the Tsai-Wu failure theory, and the optimization problem, is introduced in Chapter 2. A FE 3D model for anisotropic

material was developed in Matlab from scratch. The model was validated against the commercial software Abaqus which was further used for mesh generation. The model behaves very well and has been tested using different benchmark tests. The optimality of the structure is determined here using the Tsai-Wu failure criterion that in practice produces objective functions that are non-convex. An evolutionary optimization method is applied, namely the genetic algorithm, to search for the global solution; and, therefore, the problem becomes expensive and many evaluations of the objective function are required. Consequently, we resort to Model Order Reduction (MOR) techniques and namely the PGD framework.

A novel encapsulated PGD approach [51] is presented in Chapter 3. The proposed approach is able to handle cases where the application of standard discretization techniques would be impractical due to its computational burden. We used an in-house package of routines developed using Matlab to obtain generalized solutions, known as computational vademecum, for composite laminates that are parameterized in the fibre orientation. We obtained the affine decomposition of the elasticity and transformation tensors to have a separated stiffness matrix that is passed as an input to the PGD solver. We also extended the code with a post-process algorithm that allows us to generate the vademecum for the failure index of the laminate and then couple it with the optimization algorithm. The failure vademecum enables us to browse the failure indices of a given problem extremely fast and as many times as we want which is particularly very useful for any multi-query application and, in our case, it is used for optimizing the fibre orientation of the composite laminates. The extremely fast evaluation of the parametrized solution once it has been obtained by PGD makes the optimization possible and efficient.

The methodology is tested through a series of numerical examples in Chapter 4. The examples have been used to assess the potentiality of the methodology presented in the thesis. The numerical simulations showed excellent results in problems with two, four, and eight parameters with a significant reduction in the computational cost compared to standard FE. We also concluded that the optimal solutions depend on the domain partitioning chosen a priori.

Following the work of Alaimo *et al.* [22], in Chapter 5 we present the coupling between the PGD and clustering techniques. The aim is to search for the best strategy for the parameterization of the domain. This has been achieved by implementing a clustering optimization algorithm in Matlab that automatically finds the best sub-division of the domain based on a mechanical measure. The algorithm uses Principal Component Analysis (PCA) for dimensionality reduction and orthogonal decomposition of the data, and uses both K-means and Ward's method for clustering the data. The results showed that the optimal clustering solutions yield better optimal fibre orientation results when compared to the ones in Chapter 4 and, hence, indicating an improvement in the methodology. Finally, we concluded the work by briefly introducing additive manufacturing and

exploring the possibility of editing the open source slicing software CuraEngine in order to achieve desired printing patterns (fibre orientation). Moreover, we validated the methodology in Chapter 5 by undergoing experimental tests comparing different partitioning strategies.

6.3 Future work

The work presented in this thesis opens up other possible lines of research which can be explored in the future. Some of these are

- ***Enhancement of the model:***

  Since only linear elasticity constitutive model is employed in this thesis, it is essential to include non-linear and thermo-mechanical models to have a wider range of applications. From the PGD point of view, it is also important to enhance the vademecum by including geometrical parameterization, load location parameter, and boundary conditions parameterization. By doing so, we would obtain a more versatile model that would enable us to solve more complex geometries that would meet the industrial needs. Another way to enhance the model is to perform a multi-scale optimization analysis of fibre orientation problem by applying the homogenization theory and coupling it with PGD.

- ***Programming languages:***

  From the programming point of view, the PGD package could be implemented using high-efficiency languages such as C/C++ and/or FORTRAN. Moreover, modern simulation applications on smartphones could be developed to make use of the fast response of the PGD vademecums.

- ***Error estimation:***

  Obtaining an error estimator of a quantity of interest to be able to accurately choose the stopping criterion for the greedy algorithm in the PGD. It would also involve investigating the coupling of the stopping criteria of the greedy algorithm with the alternated directions algorithm.

- ***Additive manufacturing:***

  The presented work also opens the door to explore the possibility of printing continuous fibres with the aim of enhancing the mechanical properties of 3D printed components by avoiding jumps between partitions. We could also explore adaptive

slicing in CuraEngine, that is the slicing process that allows the variation of layer thickness according to the curvature of the model. The adaptive approach would minimize the staircase effect and the printing time [96].

Appendix A

COLLECTION OF TERMS OF THE ELASTICITY TENSOR

The collection of components process of the rotated elasticity tensor $\boldsymbol{C}(\theta)$ is performed by hand and with the aid of the symbolic tool of Matlab®. We will denote $\cos(\theta)$ as c and $\sin(\theta)$ as s for ease of notation. The components are as follows for a transversely isotropic material:

$$C_{11} = \frac{1-\nu_{23}\nu_{32}}{E_2^2\Delta}\frac{c^4}{(c^2+s^2)^2} + \frac{1-\nu_{13}\nu_{31}}{E_1E_2\Delta}\frac{s^4}{(c^2+s^2)^2} + 2\frac{\nu_{21}+\nu_{23}\nu_{31}+2E_2^2G_{12}\Delta}{E_2^2\Delta}\frac{c^2s^2}{(c^2+s^2)^2}$$

$$C_{12} = C_{21} = \frac{\nu_{21}+\nu_{23}\nu_{31}}{E_2^2\Delta}\frac{c^4}{(c^2+s^2)^2} + \frac{\nu_{21}+\nu_{23}\nu_{31}}{E_2^2\Delta}\frac{s^4}{(c^2+s^2)^2} + \frac{E_1(1-\nu_{23}\nu_{32})+E_2(1-\nu_{13}\nu_{31})-4E_1E_2^2G_{12}\Delta}{E_1E_2^2\Delta}\frac{c^2s^2}{(c^2+s^2)^2}$$

$$C_{13} = C_{31} = \frac{\nu_{31}-\nu_{21}\nu_{32}}{E_2^2\Delta}c^2 + \frac{\nu_{32}-\nu_{12}\nu_{31}}{E_1E_2\Delta}s^2$$

$$C_{16} = C_{61} = \frac{1-\nu_{21}-\nu_{23}\nu_{31}-\nu_{23}\nu_{32}-2E_2^2G_{12}\Delta}{E_2^2\Delta}\frac{c^3s}{(c^2+s^2)^2} + \frac{-E_2+E_1\nu_{21}+E_2\nu_{13}\nu_{31}+E_1\nu_{23}\nu_{31}+2E_1E_2^2G_{12}\Delta}{E_1E_2^2\Delta}\frac{cs^3}{(c^2+s^2)^2}$$

$$C_{22} = \frac{1-\nu_{13}\nu_{31}}{E_1E_2\Delta}\frac{c^4}{(c^2+s^2)^2} + \frac{1-\nu_{23}\nu_{32}}{E_2^2\Delta}\frac{s^4}{(c^2+s^2)^2} + \frac{2E_1\nu_{21}+2E_1\nu_{23}\nu_{31}+4E_1E_2^2G_{12}\Delta}{E_1E_2^2\Delta}\frac{c^2s^2}{(c^2+s^2)^2}$$

$$C_{23} = C_{32} = \frac{\nu_{31}-\nu_{21}\nu_{32}}{E_2^2\Delta}s^2 + \frac{\nu_{32}+\nu_{12}\nu_{31}}{E_1E_2\Delta}c^2$$

$$C_{26} = C_{62} = \frac{-E_2+E_1\nu_{21}+E_2\nu_{13}\nu_{31}+E_1\nu_{23}\nu_{31}+2E_1E_2^2G_{12}\Delta}{E_1E_2^2\Delta}\frac{c^3s}{(c^2+s^2)^2} + \frac{E_1-E_1\nu_{21}-E_1\nu_{23}\nu_{31}-E_1\nu_{23}\nu_{32}-2E_1E_2^2G_{12}\Delta}{E_1E_2^2\Delta}\frac{cs^3}{(c^2+s^2)^2}$$

$$C_{33} = \frac{1-\nu_{12}\nu_{21}}{E_1E_2\Delta}$$

$$C_{36} = C_{63} = \frac{-E_2(\nu_{32}+\nu_{12}\nu_{31}) + E_1(\nu_{31}-\nu_{21}\nu_{32})}{E_1E_2^2\Delta}\frac{cs}{(c^2+s^2)^2}$$

$$C_{44} = G_{23}\frac{c^2}{(c^2+s^2)^2} + G_{13}\frac{s^2}{(c^2+s^2)^2}$$

$$C_{45} = C_{54} = (G_{13}-G_{23})\frac{cs}{(c^2+s^2)^2}$$

$$C_{55} = G_{13}\frac{c^2}{(c^2+s^2)^2} + G_{23}\frac{s^2}{(c^2+s^2)^2}$$

$$C_{66} = \frac{E_2-E_1\nu_{21}+E_1-E_2\nu_{13}\nu_{31}-2E_1\nu_{23}\nu_{31}-2E_1\nu_{23}\nu_{32}-2E_1E_2^2G_{12}\Delta)}{E_1E_2^2\Delta}\frac{c^2s^2}{(c^2+s^2)^2} + G_{12}\frac{c^4}{(c^2+s^2)^2} + G_{12}\frac{s^4}{(c^2+s^2)^2}$$

$$C_{14} = C_{15} = C_{24} = C_{25} = C_{34} = C_{35} = C_{46} = C_{56} = 0$$

with

$$\Delta = \frac{1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{13}\nu_{31}-2\nu_{21}\nu_{32}\nu_{31}}{E_1E_2^2}$$

Appendix B

SEPARATION TERMS OF THE ELASTICITY AND TRANSFORMATION TENSORS

The separation of $\boldsymbol{C}$ results in the following summation for a given θ_i

$$\boldsymbol{C}(\theta_i) = \sum_{t=1}^{9} \boldsymbol{C}^t \phi_i^t(\theta_i)$$

This expression expanded has 9 terms and has the following form

$$\begin{aligned}\boldsymbol{C}(\theta_i) =& \boldsymbol{C}^1\phi_i^1(\theta_i) + \boldsymbol{C}^2\phi_i^2(\theta_i) + \boldsymbol{C}^3\phi_i^3(\theta_i) + \boldsymbol{C}^4\phi_i^4(\theta_i) + ... \\ & \boldsymbol{C}^5\phi_i^5(\theta_i) + \boldsymbol{C}^6\phi_i^6(\theta_i) + \boldsymbol{C}^7\phi_i^7(\theta_i) + \boldsymbol{C}^8\phi_i^8(\theta_i) + \boldsymbol{C}^9\phi_i^9(\theta_i)\end{aligned}$$

The separation process of $\boldsymbol{C}$ to obtain the spatial terms and the parametric terms apart was carried out by hand with the aid of the symbolic tool of Matlab®. In the following we show each spatial term with its parametric function

$$\boldsymbol{C}^1 = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 \\ & a_{22} & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & a_{66} \end{bmatrix} \qquad \phi^1(\theta_i) = \cos^4(\theta_i)$$

$$\boldsymbol{C}^2 = \begin{bmatrix} b_{11} & b_{12} & 0 & 0 & 0 & 0 \\ & b_{22} & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & b_{66} \end{bmatrix} \qquad \phi^2(\theta_i) = \sin^4(\theta_i)$$

$$\boldsymbol{C}^3 = \begin{bmatrix} d_{11} & d_{12} & 0 & 0 & 0 & 0 \\ & d_{22} & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & d_{66} \end{bmatrix} \qquad \phi^3(\theta_i) = \cos^2(\theta_i)\sin^2(\theta_i)$$

$$\boldsymbol{C}^4 = \begin{bmatrix} 0 & 0 & e_{13} & 0 & 0 & 0 \\ & 0 & e_{23} & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & e_{44} & 0 & 0 \\ & & & & e_{55} & 0 \\ & & & & & 0 \end{bmatrix} \qquad \phi^4(\theta_i) = \cos^2(\theta_i)$$

$$\boldsymbol{C}^5 = \begin{bmatrix} 0 & 0 & g_{13} & 0 & 0 & 0 \\ & 0 & g_{23} & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & g_{44} & 0 & 0 \\ & & & & g_{55} & 0 \\ & & & & & 0 \end{bmatrix} \qquad \phi^5(\theta_i) = \sin^2(\theta_i)$$

$$\boldsymbol{C}^6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & h_{16} \\ & 0 & 0 & 0 & 0 & h_{26} \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad \phi^6(\theta_i) = \cos^3(\theta_i)\sin(\theta_i)$$

$$\boldsymbol{C}^7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & k_{16} \\ & 0 & 0 & 0 & 0 & k_{26} \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad \phi^7(\theta_i) = \cos(\theta_i)\sin^3(\theta_i)$$

$$\boldsymbol{C}^8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & l_{36} \\ & Sym. & & 0 & l_{45} & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad \phi^8(\theta_i) = \cos(\theta_i)\sin(\theta_i)$$

$$\boldsymbol{C}^9 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & q_{33} & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad \phi^9(\theta_i) = 1$$

Note that all the components $a, b, d, e, g, h, k, l, q$ are function of the material characteristics, i.e. Young's moduli, shear moduli, and Poisson's ratios, for transversely isotropic material.

Similarly, the separation of the transformation matrix $\boldsymbol{T}$ results in the following summation for a given θ_i

$$\boldsymbol{T}(\theta_i) = \sum_{r=1}^{7} \boldsymbol{T}^r Z_i^r(\theta_i)$$

This expression expanded has 7 terms and has the following form

$$\begin{aligned} \boldsymbol{T}(\theta_i) = & \boldsymbol{T}^1 Z_i^1(\theta_i) + \boldsymbol{T}^2 Z_i^2(\theta_i) + \boldsymbol{T}^3 Z_i^3(\theta_i) + ... \\ & \boldsymbol{T}^4 Z_i^4(\theta_i) + \boldsymbol{T}^5 Z_i^5(\theta_i) + \boldsymbol{T}^6 Z_i^6(\theta_i) + \boldsymbol{T}^7 Z_i^7(\theta_i) \end{aligned}$$

It is shown next the spatial parts of $\boldsymbol{T}$ each with its parametric function

$$\boldsymbol{T}^1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad Z^1(\theta_i) = \cos^2(\theta_i)$$

$$\boldsymbol{T}^2 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad Z^2(\theta_i) = \sin^2(\theta_i)$$

$$\boldsymbol{T}^3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & Sym. & & 1 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad Z^3(\theta_i) = \cos(\theta_i)$$

$$\boldsymbol{T}^4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad Z^4(\theta_i) = \sin(\theta_i)$$

$$\boldsymbol{T}^5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad Z^5(\theta_i) = \cos(\theta_i)\sin(\theta_i)$$

$$\boldsymbol{T}^6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 1 \end{bmatrix} \qquad Z^6(\theta_i) = \cos^2(\theta_i) - \sin^2(\theta_i)$$

$$\boldsymbol{T}^7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & Sym. & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \qquad Z^7(\theta_i) = 1$$

Appendix C

MATERIAL CHARACTERISTICS OF CARBON FIBRE REINFORCED ABS

The elasticity tensor is described by characteristic values of the material such as Young's modulii, Poison's ratios, and shear modulii. The elasticity tensor has the following format for orthotropic materials,

$$\begin{bmatrix} \frac{1-\nu_{23}\nu_{32}}{E_2E_3\Delta} & \frac{\nu_{21}-\nu_{23}\nu_{31}}{E_2E_3\Delta} & \frac{\nu_{31}-\nu_{21}\nu_{32}}{E_2E_3\Delta} & 0 & 0 & 0 \\ & \frac{1-\nu_{13}\nu_{31}}{E_1E_3\Delta} & \frac{\nu_{32}-\nu_{12}\nu_{31}}{E_1E_3\Delta} & 0 & 0 & 0 \\ & & \frac{1-\nu_{12}\nu_{21}}{E_1E_2\Delta} & 0 & 0 & 0 \\ & \text{Sym.} & & G_{23} & 0 & 0 \\ & & & & G_{13} & 0 \\ & & & & & G_{12} \end{bmatrix}$$

where,

$$\Delta = \frac{1-\nu_{12}\nu_{21}-\nu_{23}\nu_{32}-\nu_{13}\nu_{31}-2\nu_{21}\nu_{32}\nu_{31}}{E_1E_2E_3}$$

Note that, for transversely isotropic material with plane 2-3 as the plane of isotropy, the following relations hold

$$E_2 = E_3, \qquad \nu_{12} = \nu_{13}, \qquad \nu_{32} = \nu_{23}, \qquad \nu_{21} = \frac{E_2}{E_1}\nu_{12}$$
$$\nu_{31} = \nu_{21}, \qquad G_{12} = G_{13}, \qquad \text{and} \qquad G_{23} = \frac{E_2}{2(1+\nu_{23})}$$

In this work, we used a material close to carbon fibre ABS material in our simulations, and the characteristics have the following values:

Material characteristics				
E_1 [MPa]	E_2 [MPa]	ν_{12}	ν_{23}	G_{12} [MPa]
5.71481×10^3	2.74085×10^3	0.164	0.38	1106.85

Table C.1: Carbon fibre reinforced ABS material characteristics

Appendix D

CLUSTERING TECHNIQUES

D.1 K-means

K-means is very popular and is by far the most widely used clustering algorithm. Given an unlabeled data set, the K-means algorithm aims to partition the data into a given number of mutually exclusive clusters K. Each observation in the data set is treated as a point having a location in space as shown in the example in Fig. D.1.

K-means algorithm is an iterative algorithm and performs two main steps [97]. The first step is cluster assignment and the second step is centroid moving. During the cluster assignment step, the algorithm is going through each of the observations in the data set, i.e. the black dots shown in Fig. D.2a; and depending on whether it's closer to the red cluster centroid or the blue cluster centroid which are randomly initialized, the algorithm is going to assign each of the data points to one of the two cluster centroids. Specifically, what is meant by that, is to go through the data set and color (assign) each of the points either in red or blue, depending on whether it is closer to the red cluster centroid or the blue cluster centroid.
The other part of the loop of K-means is the move centroid step. We take the two cluster centroids, that is, the red cross and the blue cross in Fig. D.2a, and we move them to the average of the points having the same color. In other words, we examine all the red points and compute the average resulting in the mean of the location of all the red points, and then we move the red cluster centroid there. Similarly, we perform the same step for the the blue cluster centroid. We look at all the blue dots and compute their

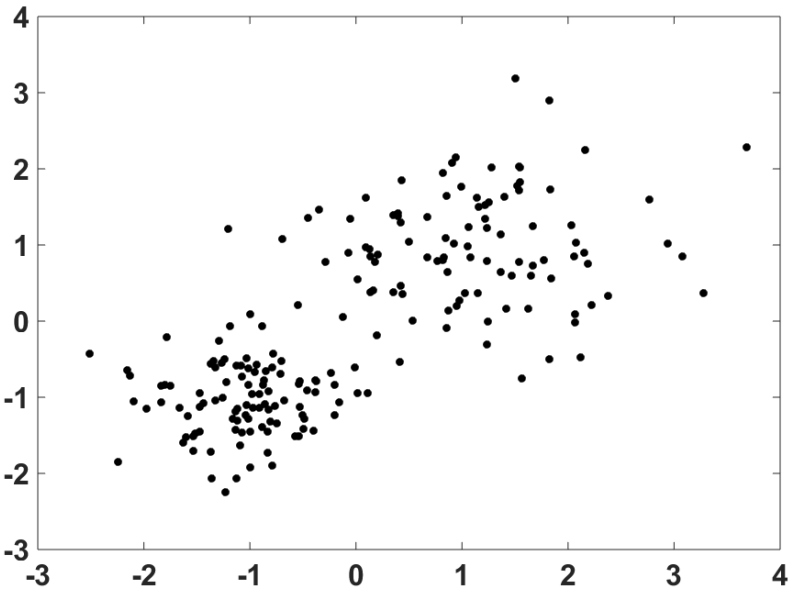


Figure D.1: Example data set

mean, and then move the blue cluster centroid there. We keep repeating those steps until the cluster centroids don't change any further. Altogether, the two main steps could be expanded into six mini-steps:

(1) Randomly initialize K cluster centroids as shown in Fig. D.2a.

(2) Calculate distances of data points from centroids.

(3) Assign each data point to the closest centroid.

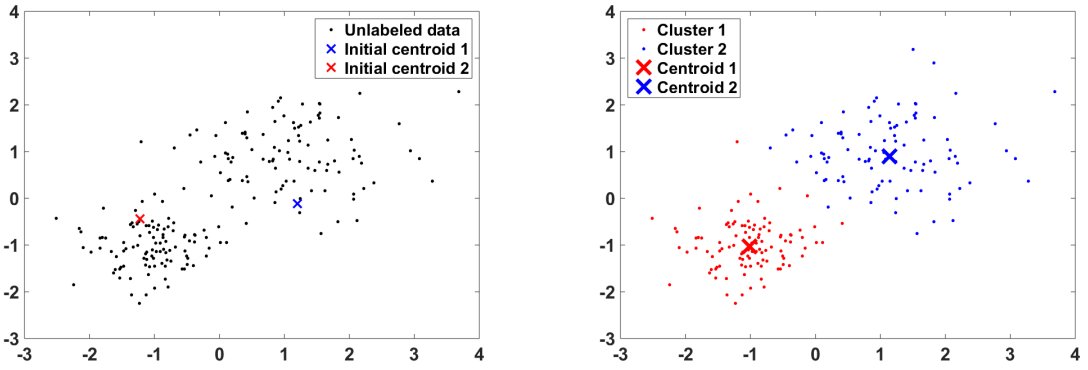
(4) Compute the mean of each cluster.

(5) Move the centroid to the location of the mean.

(6) Repeat steps 2-5 until convergence (shown in Fig. D.2b).

Note that the distances calculated in step (2) often do not represent spatial distances.

The K-means algorithm has an optimization objective or a cost function $\mathcal{J}$ that it is trying to minimize. It is very useful to show it here to deeply get the idea behind the algorithm. Therefore, understanding what is the optimization objective of K-means will help us to debug the algorithm and just make sure that K-means is running correctly and, more importantly, we will also be able to avoid local optima.



(a) Randomly initialized centroids for $K = 2$

(b) Cluster assignments and centroids for $K = 2$

Figure D.2: Initial and final centroids positions and cluster solution for an example with $K = 2$

Let us consider two sets of variables. First is the $c^{(i)}$ and it keeps track of the index or the number of the cluster to which an observation from the data set $x^{(i)}$ is assigned. The superscript i denotes the i-th observable in the data set and it runs from $1, \ldots, m$, where m is the total number of observations in the data set. The other set of variables is ψ_k, which is the location of the cluster centroid k. As mentioned before, for K-means we use capital K to denote the total number of clusters and lower case k is going to be an index into the cluster centroids running from $1, \ldots, K$.

A variation of ψ_k is the $\psi_{c^{(i)}}$ which denotes the centroid location of the cluster to which example $x^{(i)}$ has been already assigned. For example let us assume that $x^{(i)}$ has been assigned to cluster number two, in this case $c^{(i)}$ that is the index of $x^{(i)}$, is equal to two and, consequently, $\psi_{c^{(i)}}$ is equal to ψ_2.

The objective function that K-means is minimizing is function of all of the sets of variables $c^{(1)}, \ldots, c^{(m)}$ and $\psi_1, \ldots, \psi_K$. With the just mentioned notation, we could now express the optimization objective of the K-means clustering algorithm as follows,

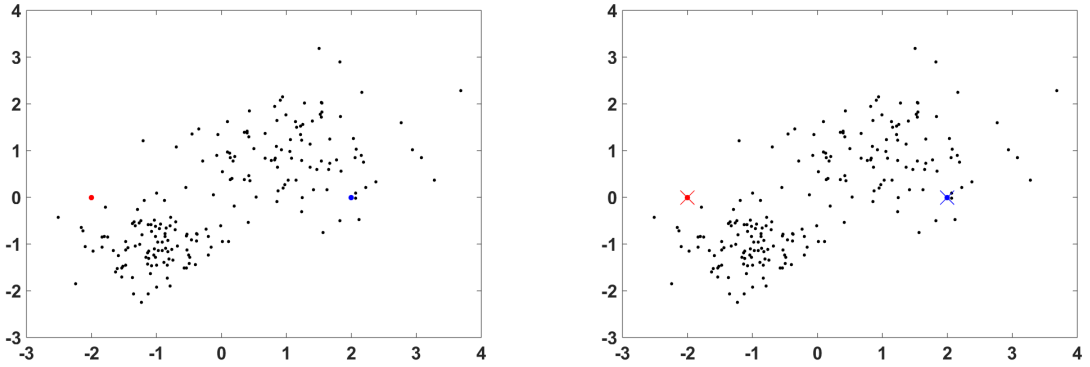
$$\mathcal{J}(c^1, \ldots, c^m, \psi_1, ..., \psi_K) = \frac{1}{m} \sum_{i=1}^{m} \|x^{(i)} - \psi_{c^{(i)}}\|^2. \tag{D.1}$$

What K-means can be shown to be doing is that it is trying to define parameters $c^{(i)}$ and $\psi_{c^{(i)}}$ to minimize the cost function $\mathcal{J}$. This cost function is sometimes also called the distortion cost function or the distortion of the K-means algorithm. In other words, the cluster assignment step is exactly minimizing $\mathcal{J}$ with respect to the variables $c^1, c^2, \ldots, c^m$ while holding the cluster centroids locations $\psi_1, \ldots, \psi_K$ fixed.

The cluster assignment step does not change the cluster centroids location, but it is exactly picking the values of $c^1, c^2, \ldots, c^m$ that minimize the cost function, or the distortion function $\mathcal{J}$. The interested reader should refer to [97] and the references therein for deeper insight and mathematical proofs. The intuitive meaning is assigning each point to a cluster centroid that is closest to it because that is what minimizes the square of distance between the points and the cluster centroids. The second step is the move centroid step. What the move centroid step does is that it chooses the values of ψ that minimizes $\mathcal{J}$, i.e., it minimizes the cost function $\mathcal{J}$ with respect to the locations of the cluster centroids $\psi_1, \ldots, \psi_K$. The minimization problem could be written as follows,

$$\arg \min_{\substack{c^1,\ldots,c^m \\ \psi_1,\ldots,\psi_K}} \mathcal{J}(c^1, \ldots, c^m, \psi_1, ..., \psi_K). \tag{D.2}$$

The minimization of (D.2) will lead to the discussion of how to make K-means avoid local optima. There are many ways of choosing the initial centroids in step (1) of the K-means cluster analysis algorithm. A poor choice of the initial centroids could result in sub-optimal clusters, meaning that the minimization of the total sum of distances between observations and centroids would not converge to a global optimum. One initialization approach is to simply choose the initial centroids randomly from among the observations in the data set. When running the K-means algorithm, we should have the number of cluster centroids, K, set to be less than the number of observations in our data set m as it does not make sense to have more centroids than the number of observations. K-means is initialized by randomly picking K observations from the data set, which is basically equal to the predefined number of cluster centroids. Then we set the centroid $\psi_1, \ldots, \psi_K$ to be in the randomly chosen K observations locations.



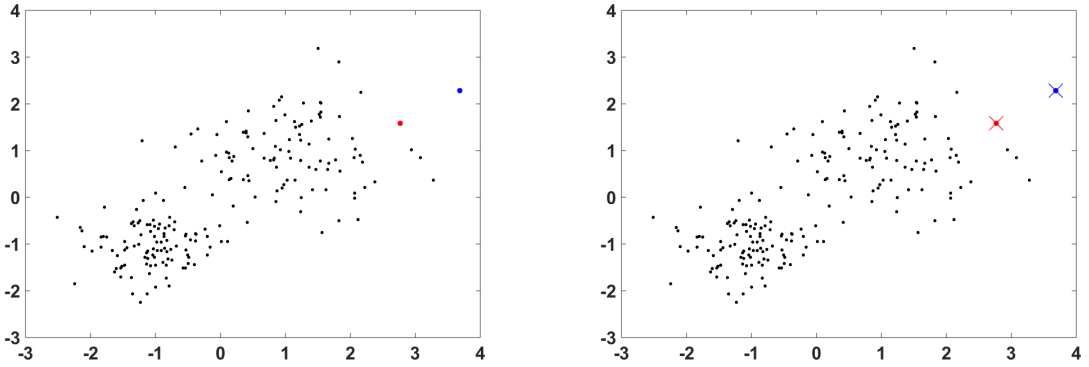
(a) Randomly picked points in the data set

(b) Centroids assignment to randomly generated points

Figure D.3: Steps for random initialization of cluster centroids for an example with $K = 2$

For example, if K is equal to two and, for the example in Fig. D.1, we would like to find two clusters. In order to initialize the cluster centroids, we randomly pick a couple of

observations (because $K = 2$ in this example) as shown in Fig. D.3a in blue and red. Consequently, we initialize the cluster centroids to be right on top of those randomly chosen points as shown in Fig. D.3b. In other words, at initialization, our first cluster centroid ψ_1 will be equal to $x^{(i)}$ for some randomly chosen value of i and ψ_2 will be equal to $x^{(j)}$ for some different randomly chosen value of j. The output of K-means for this example is shown in Fig. D.2b. Note that this initialization was particularly a good one since the centroids are quite distant and well positioned amid the data points.



(a) Randomly picked points in the data set

(b) Centroids assignment to randomly generated points

Figure D.4: Bad initialization of cluster centroids for an example with $K = 2$

However, sometimes we could get less lucky and maybe we end up picking points that are close to each other as our initial random starting observation as shown in Fig. D.4a. This results in being stuck in a local minimum when minimizing the cost function (D.1). We can easily observe that the centroids are not placed in a good balanced location amid the data points anymore in Fig. D.4b. As a result, the K-means clustering algorithm yields a wrong solution as shown in Fig. D.5. It seems that in Fig. D.5 the red cluster has captured almost all of the points in the data set while the blue cluster captured only one observation, and therefore, corresponding to a bad local optimum.

By comparing the two solutions in Fig. D.2b and Fig. D.5, we conclude that K-means could end up converging to different solutions depending on exactly how the clusters were initialized, and therefore, affected by the "random" initialization. In order to alleviate the problem of K-means getting stuck in local optima and increase the odds of K-means finding the best possible clustering, one could try multiple random initialization. Meaning that, instead of just initializing K-means once and running the algorithm just once, we initialize K-means and run the whole algorithm several times. By doing so, we ensure that we have different random initialization and, therefore, leading to better solutions, possibly a global optimum.

The multiple initialization is simple to perform; let us assume that we decide to initialize and run K-means for 200 times (50-1000 times typically). Thus, the outcome of the

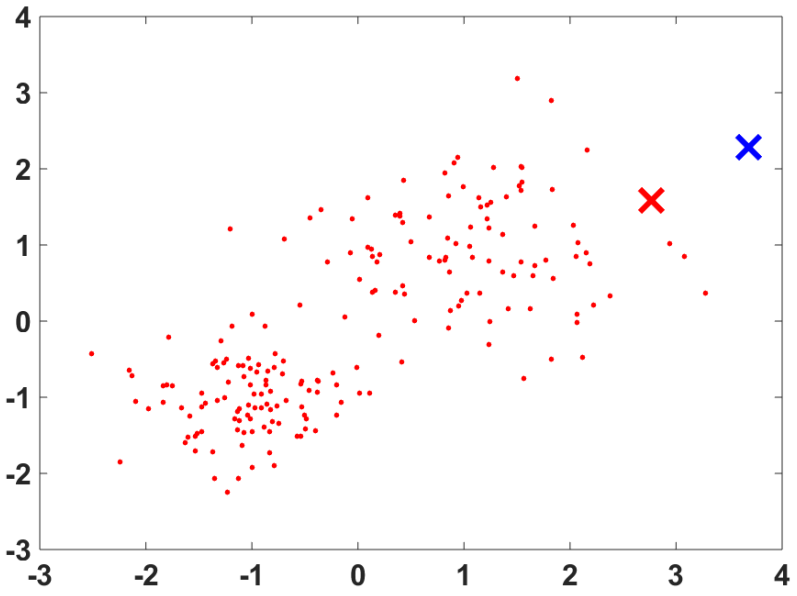


Figure D.5: Local minimum solution from the minimization of the K-means objective function

K-means would give us a set of clustering solutions and a set of cluster centroids. We would then compute the cost $\mathcal{J}$ of the set of cluster assignments and cluster centroids that we got. Finally, having done this whole procedure 200 times, we will have 200 different ways of clustering the data and, ultimately, we just pick the one that gives us the lowest cost $\mathcal{J}$.

By experiment, we could also deduce that if we are running K-means with a fairly small number of clusters, with the number of clusters anywhere between two to ten, then doing multiple random initialization can often ensure that we find a better local optimum. However, if K is very large (hundreds), then, having multiple random initialization is less likely to make a huge difference and there is a much higher chance that our first random initialization will yield a decent solution.

Another approach is to choose one centroid randomly and then choose the others so as to try to spread the centroids out so that they are as far apart from each other as possible. The just mentioned approach is called the K-means++ algorithm. The K-means++ algorithm is a variation of the K-means algorithm. It initializes the centroids based on probabilities. The algorithm is widely used and is built-in in most programming libraries. The K-means++ algorithm has the same steps as the K-means algorithm except that step (1) in K-means is expanded into the following four steps:

(a) Choosing a random observation $x^{(i)}$ from the data set to be the first centroid ψ_1

(b) Compute the distance $D(x)$ between the observation and the nearest centroid that is already chosen

(c) Randomly choose a new observation as a new centroid according the a weighted probability distribution that ensures the spread of centroids

(d) Finally repeat steps (b) and (c) until we have chosen K centroids

It was reported in [98] that the K-means++ algorithm showed excellent capabilities with real big data sets. The algorithm shows high accuracy while maintaining a decent speed.

D.2 Hierarchical clustering

Another popular type of clustering techniques is the hierarchical clustering. The basic idea of the hierarchical clustering is to produce a set of clusters and establish the relationships between them [99]. The idea is to build a tree of clusters that have different levels (or hierarchy). The tree, also called *dendrogram*, is not a single set of clusters but rather a multilevel hierarchy where clusters at one level are then joined together or split in the following level. In other words, the tree that is built is the whole possible clustering solutions as shown in the example in Fig. D.6.

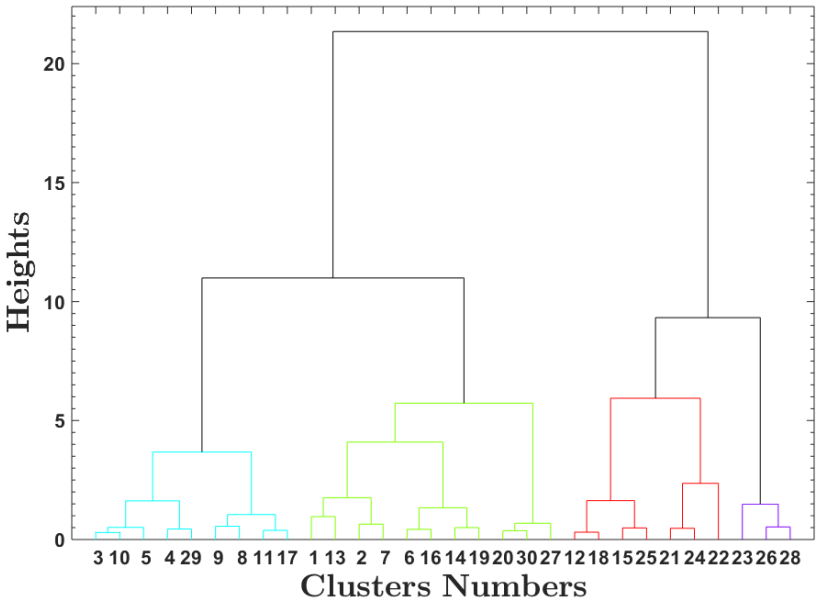


Figure D.6: Cluster tree or dendrogram example

Note that unlike K-means, a single observation of the data set could belong to more than one cluster as long as these clusters are related and have different hierarchies or levels. The hierarchy allows us to decide which level or height of clustering is suitable for our application. There are two main types of hierarchical clustering [20]. The first is the agglomerative hierarchical clustering and the second is the divisive hierarchical clustering. They are essentially following the same idea but with different tree building direction. The agglomerative hierarchical clustering constructs its clusters in a bottom-up fashion. Meaning that, each observation in the data set starts off in its own cluster as a singleton, leading to as many clusters as observations in the first hierarchy (level). The clusters are then merged together, in a binary mode, based on a given distance measure (also called linkage). The linking process keeps going in an iterative way until all the clusters are joined into one cluster (the root of the tree). Whereas the divisive clustering technique is a top-bottom approach. It initially starts with one cluster, that is the root of the tree, and then it is successively divided into sub-clusters and so on recursively until the tree nodes are obtained (leaves of the tree). The result of either technique is the dendrogram that represents the nested groups of clustering and showing which cluster merged (or split) with which cluster highlighting the similarity levels at which groups of clusters change as shown in Fig. D.6. We will explain a little bit more in detail the agglomerative hierarchical clustering as it is commonly used in different applications and we will employ it in our analyses.

As mentioned before, the grouping of clusters (or the split of clusters in divisive clustering) occurs according to a similarity measure between observations in a data set [100]. This is achieved first by computing the distances between observations using a metric measure , such as Euclidean distance, then the similarity is found by using a linkage criterion between sets of clusters. The linkage criterion obtains the distances between sets of observations as a function of the pairwise distances computed between observations [100]. Let us assume that the distance between any two observations of a set S is denoted as $D(x^{(i)}, x^{(j)})$. To be able to select the closest pair of clusters to merge together at each stage of the process, we need to define a sub-set distance $\Delta(S_i, S_j)$ that is the pairwise distance between elements belonging to different sub-sets. The sub-sets are equal to $S_i = \{x^{(i)}\}$ and $S_j = \{x^{(j)}\}$ only when S_i and S_j are singletons, which is the first stage of agglomerative clustering; and in this case $D(x^{(i)}, x^{(j)}) = \Delta(S_i, S_j)$ [100]. The linkage function could be divided into three main categories as follows:

(1) **Single linkage:** also called the nearest neighbour method and it defines the distance between two clusters as the shortest distance between any element of one cluster to any element of the other cluster.

$$\Delta(S_i, S_j) = \min_{\substack{x^{(i)} \in S_i \\ x^{(j)} \in S_j}} D(x^{(i)}, x^{(j)}). \tag{D.3}$$

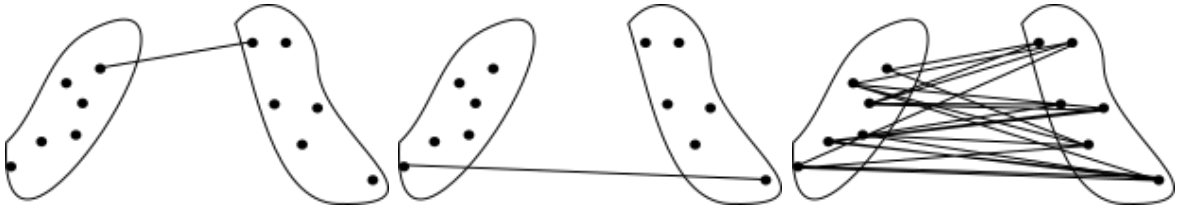
(2) **Complete linkage:** also called the furthest neighbour method and it defines the distance between two clusters as the longest distance between any element of one cluster to any element of the other cluster.

$$\Delta(S_i, S_j) = \max_{\substack{x^{(i)} \in S_i \\ x^{(j)} \in S_j}} D(x^{(i)}, x^{(j)}). \tag{D.4}$$

(3) **Average linkage:** also called minimum variance method and it defines the distance between two clusters as the average of all the distances between elements of one cluster and elements of the other cluster.

$$\Delta(S_i, S_j) = \frac{1}{|S_i||S_j|} \sum_{x^{(i)} \in S_i} \sum_{x^{(j)} \in S_j} D(x^{(i)}, x^{(j)}). \tag{D.5}$$

The linkage functions could be also illustrated as shown in Fig. D.7 for clarity.



(a) Single linkage (b) Complete linkage (c) Average linkage

Figure D.7: Types of linkage distances in hierarchical clustering

One of the drawbacks of hierarchical clustering is that it requires a lot of memory and it is time consuming for big data [100]. It was also reported that the algorithm with complete linkage (also called CLINK) is very sensitive to outliers in the data. While the algorithm with single linkage function (also called SLINK) yields an unwanted "*chaining phenomenon*" in dendrograms, which is the event where two clusters are joined because there are two elements very close to each other while the other elements are very distant. This issue is alleviated in the complete linkage as it ensures to merge clusters having similar diameters [101].

There exists more sub-set linkage distances in the literature such as weighted average linkage, centroid linkage, and Ward's linkage. Ward's method is widely used as it possesses both cluster homogeneity and cluster separability. The method aims to minimize the increase in the total within-cluster sum of squared error, that is, the total within-cluster variance of data [102]. In its original format, it uses the squared Euclidean distance as the metric for distances between observations in the data. The method is widely used due to its popularity, and it has been extended into several versions [99]. Ward's method merges in an iterative way two clusters at a time ensuring the minimum within-cluster

variance possible [103]. The algorithm is very straightforward and it has four main steps as follows:

(1) Initialize for each observation in the data set to be assigned to its cluster singleton $S_i = \{x^{(i)}\}$ and put them in a list. The set $S = \{S_1, S_2, \ldots, S_K\}$ where K is the total number of clusters (and observations).
(2) The pair S_i and S_j are grouped together based on the linkage function yielding a new cluster $S_{ij} = S_i \cup S_j$. The old sub-sets S_i and S_j are then removed from the list.
(3) Compute and set the centroid of the newly obtained cluster to its center of gravity.
(4) Finally, the total number of clusters K reduces by one. If K is larger than the desired number of clusters, go back to step (2).

The linkage distance in Ward's method is measured as follows,

$$\Delta(S_i, S_j) = \frac{n_i n_j}{n_i + n_j} D(\psi_i, \psi_j), \tag{D.6}$$

where n_i and n_j are the number of observations in sub-sets S_i and S_j respectively; $D(\psi_i, \psi_j)$ is a metric measure and in Ward's method's original form it is the squared Euclidean distance, $D(\psi_i, \psi_j) = \|\psi_i - \psi_j\|^2$, between centroids ψ_i and ψ_j. In the following section, we will show how these clustering techniques play a role in the optimization of fibre orientation in composite laminates using the PGD framework.

Appendix E

MATHEMATICS OF PRINCIPAL COMPONENTS

The simplest way to derive the principal components could be achieved by finding the projections which maximize the variance of the data. The first principal component is the direction in space along which projections have the largest variance.

Let us assume we have the $m \times p$ data matrix $\boldsymbol{X}$, where m is the number of observations in the data and p is the number of snapshots taken of the system (also called number of features). We will also assume that the data is centered. We are now looking for a vector $\boldsymbol{v}$ that maximizes the variance of the data. In order to derive the first principal component we need to define first the variance. Given the covariance matrix $\boldsymbol{\Sigma} = \dfrac{\boldsymbol{X}^\mathsf{T}\boldsymbol{X}}{m}$, the variance could be written as

$$\mathrm{Var}(X) = \sigma^2 = v^\mathsf{T} \Sigma v$$

We would like to find a vector $\boldsymbol{v}$, having a unit length $\boldsymbol{v}^\mathsf{T}\boldsymbol{v} = 1$ to ensure the well posedness of the problem, that maximizes $\boldsymbol{v}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{v}$. The constrained optimization problem could be transformed into an unconstrained optimization problem by introducing a Lagrange multiplier λ. The problem as a result could be seen as maximizing $\boldsymbol{v}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{v} - \lambda(\boldsymbol{v}^\mathsf{T}\boldsymbol{v} - 1)$. We then differentiate the objective function with respect to the vector $\boldsymbol{v}$ and setting the derivative to zero to maximize, yielding the following eigenvector equation

$$\boldsymbol{\Sigma}\boldsymbol{v} - \lambda\boldsymbol{v} = 0$$

where $\boldsymbol{v}$ is an eigenvector of the covariance matrix $\boldsymbol{\Sigma}$, and thus the vector yielding the maximum variance will be the one corresponding to the largest eigenvalue λ.

Appendix F

FACTORS OF THE RAW SAFETY FACTOR DATA AND ITS CLUSTERING

In this section we provide complementary figures for the analysis and the clustering of factors obtained from applying PCA on the safety factor raw data $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}}$ presented in Chapter 5.

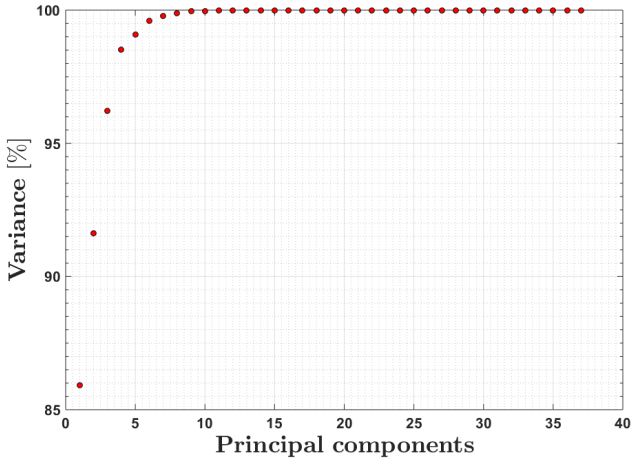
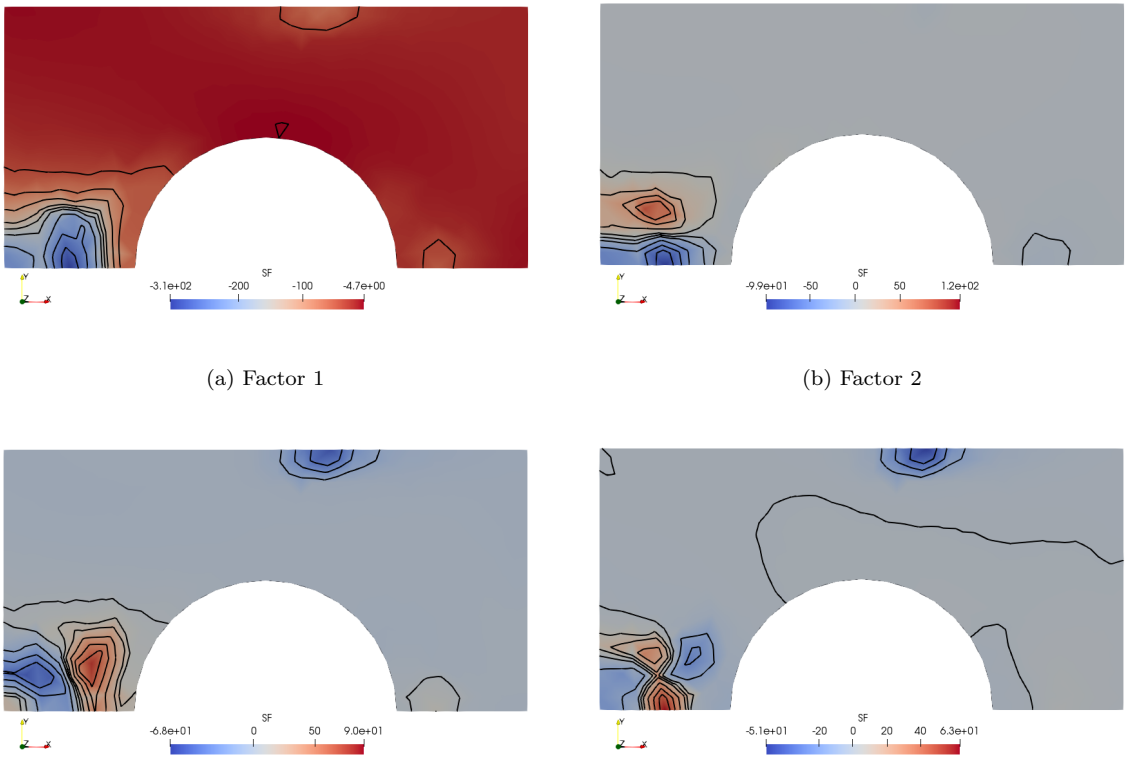


Figure F.1: Evolution of the cumulative variance with the number of factors for the raw safety factor data



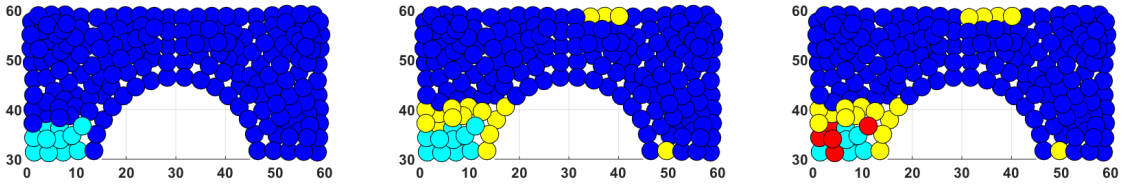
(a) Factor 1

(b) Factor 2

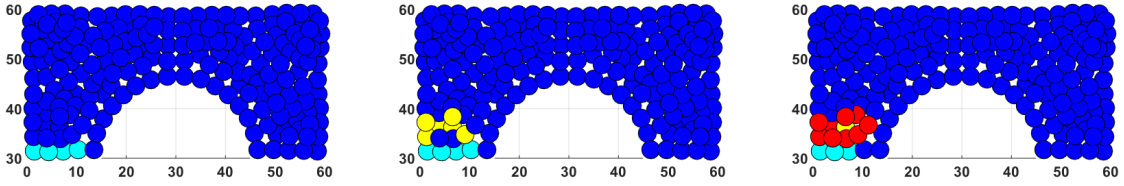
(c) Factor 3

(d) Factor 4

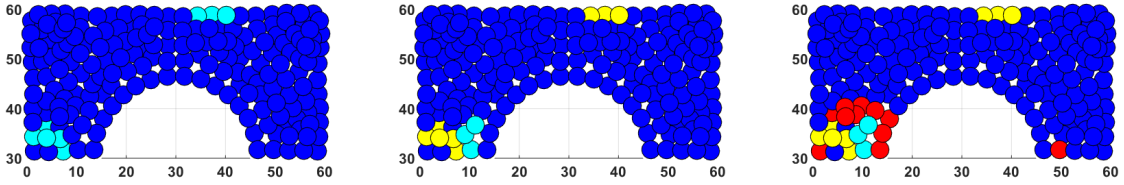
Figure F.2: The first four factors obtained from applying PCA on the safety factor raw data matrix $\tilde{\boldsymbol{\lambda}}_{\boldsymbol{s}}$



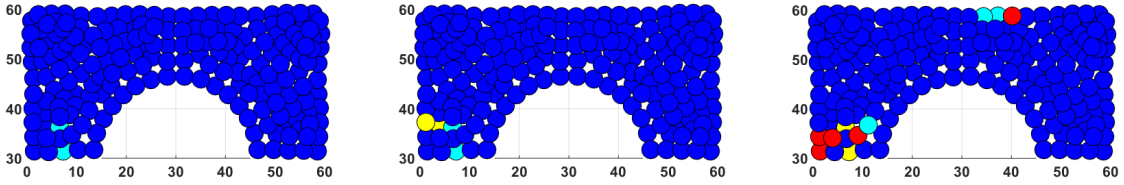
(a) Factor 1



(b) Factor 2

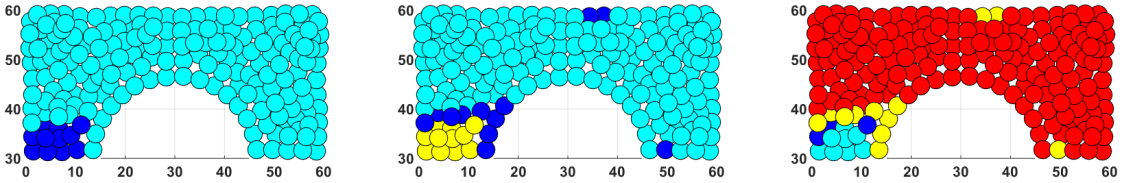


(c) Factor 3

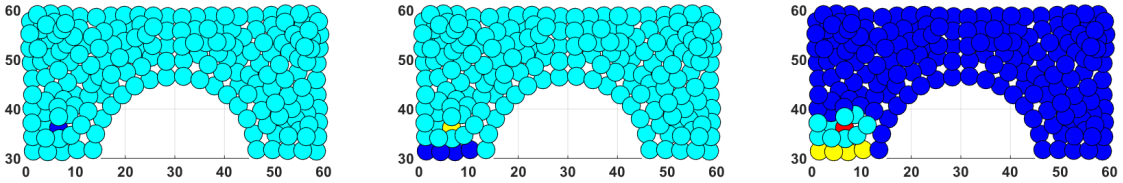


(d) Factor 4

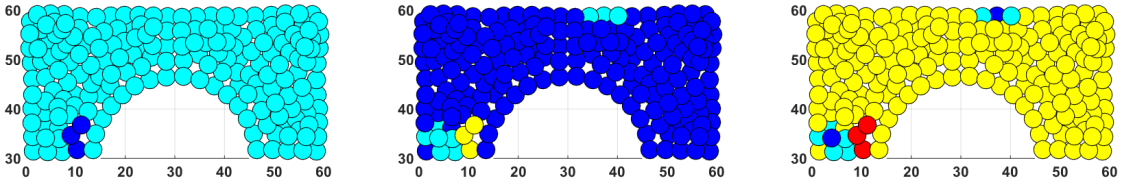
Figure F.3: Clustering of the first four factors obtained from applying PCA on the safety factor raw data matrix using K-means. Each factor is divided into 2,3,4 clusters from left to right.



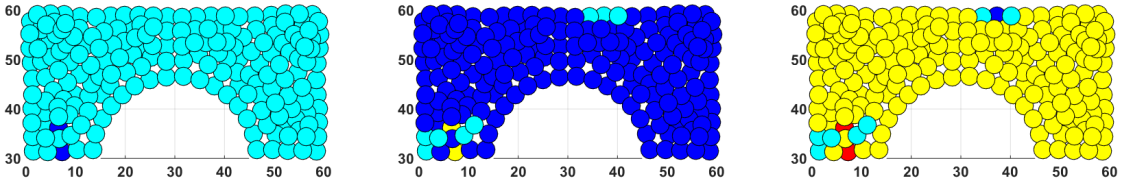
(a) Factor 1



(b) Factor 2



(c) Factor 3



(d) Factor 4

Figure F.4: Clustering of the first four factors obtained from applying PCA on the safety factor raw data matrix using Ward's method. Each factor is divided into 2,3,4 clusters from left to right.

BIBLIOGRAPHY

[1] Ekkehard Ramm, Kurt Maute, and Stefan Schwarz. Conceptual design by structural optimization. *Proceedings of the Euro–C 1998 Conference on Computational Modelling of Concrete Structures, herausgegeben von R. de Borst, N. Bicanic, H. Mang & G. Meschke, S*, pages 879–896, 1998.

[2] Hongmei Gu and John F. Hunt. Two-dimensional finite element heat transfer model of softwood. Part III. Effect of moisture content on thermal conductivity. *Wood and Fiber Science*, 39(1):159–166, 2007. ISSN 07356161.

[3] Tomás Zegard and Glaucio H. Paulino. Bridging topology optimization and additive manufacturing. *Structural and Multidisciplinary Optimization*, 53(1):175–192, 2016. ISSN 16151488. doi: 10.1007/s00158-015-1274-4.

[4] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, and A. Poitou. Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity. *Computer Methods in Applied Mechanics and Engineering*, 201-204:1–12, 2012. ISSN 00457825. doi: 10.1016/j.cma.2011.08.025.

[5] Francisco Chinesta, Roland Keunings, and Adrien Leygue. *The proper generalized decomposition for advanced numerical simulations: A primer*. Number 9783319028644. 2014. ISBN 9783319028644. doi: 10.1007/978-3-319-02865-1.

[6] Haichao An, Shenyan Chen, and Hai Huang. Stacking sequence optimization and blending design of laminated composite structures. *Structural and Multidisciplinary Optimization*, 59(1):1–19, 2019. ISSN 16151488. doi: 10.1007/s00158-018-2158-1.

[7] Narindra Ranaivomiarana. *Simultaneous optimization of topology and material anisotropy for aeronautic structures*. PhD thesis, SORBONNE UNIVERSITE, 2019.

[8] S. Nikbakt, S. Kamarian, and M. Shakeri. A review on optimization of composite structures Part I: Laminated composites. *Composite Structures*, 195(November 2017):158–185, 2018. ISSN 02638223. doi: 10.1016/j.compstruct.2018.03.063. URL https://doi.org/10.1016/j.compstruct.2018.03.063.

[9] F. Chinesta, A. Ammar, A. Leygue, and R. Keunings. An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11):578–592, 2011. ISSN 03770257. doi: 10.1016/j.jnnfm.2010.12.012.

[10] Carl Zweben. Composite Materials. In *Comprehensive Composite Materials*, chapter 10. Devon, 2000. doi: 10.2307/25304499.

[11] Tri-Dung Ngo. Introduction to Composite Materials. *Fiber Composites [Working Title]*, 2020. doi: 10.5772/intechopen.91285.

[12] Andre Veldmeijer. UCLA Encyclopedia of Egyptology. *Deir el-Gabrawi*, pages 1–5, 2012. URL syncii:///Deirel-Gabrawi.pdf.

[13] Myer Kutz. *Mechanical Engineer's Handbook*. 2015. ISBN 978-1-118-90748-1.

[14] K. Balasubramanian, Mohamed T.H. Sultan, and N. Rajeswari. *Manufacturing techniques of composites for aerospace applications*. Elsevier Ltd, 2018. ISBN 9780081021316. doi: 10.1016/B978-0-08-102131-6.00004-9. URL http://dx.doi.org/10.1016/B978-0-08-102131-6.00004-9.

[15] Peter W. Christensen and Anders Klarbring. *An Introduction to Structural Optimization*, volume 153. 2008. ISBN 9788578110796. doi: 10.1017/CBO9781107415324.004. URL http://books.google.com/books?hl=en{&}lr={&}id=0qd6vd12sXEC{&}oi=fnd{&}pg=PA2{&}dq=an+introduction+to+structural+optimization{&}ots=RGcjsA-gt{_}{&}sig=b7mV-Kil1ibHPWvkJ6stdU1{_}3EE{%}5Cnhttp://link.springer.com/10.1007/978-1-4020-8666-3.

[16] G.I.N Rozvany. *Optimization of Large Structural Systems*. Springer-science, 1993. ISBN 9789401095792. doi: 10.1007/978-94-010-9577-8_9.

[17] Alberto Pedro Sibileau. *Computational Vademecums for Lattice Materials using algebraic PGD by*. PhD thesis, UPC, 2019.

[18] Rubén Ibáñez, Emmanuelle Abisset-Chavanne, Francisco Chinesta, Antonio Huerta, and Elías Cueto. A local multiple proper generalized decomposition based on the partition of unity. *International Journal for Numerical Methods in Engineering*, 120(2):139–152, 2019. ISSN 10970207. doi: 10.1002/nme.6128.

[19] E. Lopez, D. Gonzalez, J. V. Aguado, E. Abisset-Chavanne, E. Cueto, C. Binetruy, and F. Chinesta. A Manifold Learning Approach for Integrated Computational Materials Engineering. *Archives of Computational Methods in Engineering*, 25(1):59–68, 2018. ISSN 18861784. doi: 10.1007/s11831-016-9172-5.

[20] O Maimon and Lior Rokach. *The Data Mining and Knowledge Discovery Handbook*. Springer, 2005. ISBN 9780387244358.

[21] Jared Dean. *Big Data Mining and Machine Learning*. Wiley, 2014. ISBN 978-1-118-92069-5.

[22] Gianluca Alaimo, Ferdinando Auricchio, Sonia Marfia, and Elio Sacco. Optimization clustering technique for PieceWise Uniform Transformation Field Analysis homogenization of viscoplastic composites. *Computational Mechanics*, 64(6):1495–1516, 2019. ISSN 14320924. doi: 10.1007/s00466-019-01730-2. URL https://doi.org/10.1007/s00466-019-01730-2.

[23] Delin Jiang, Robert Hoglund, and Douglas E. Smith. Continuous fiber angle topology optimization for polymer composite deposition additive manufacturing applications. *Fibers*, 7(2), 2019. ISSN 20796439. doi: 10.3390/FIB7020014.

[24] J. Huang and R. T. Haftka. Optimization of fiber orientations near a hole for increased load-carrying capacity of composite laminates. *Structural and Multidisciplinary Optimization*, 30(5):335–341, 2005. ISSN 1615147X. doi: 10.1007/s00158-005-0519-z.

[25] J. H. Luo and H. C. Gea. Optimal orientation of orthotropic materials using an energy based method. *Structural Optimization*, 15(3-4):230–236, 1998. ISSN 09344373. doi: 10.1007/BF01203536.

[26] P. Pedersen. Bounds on elastic energy in solids of orthotropic materials. *Structural Optimization*, 2(1):55–63, 1990. ISSN 09344373. doi: 10.1007/BF01743521.

[27] P. Pedersen. On optimal orientation of orthotropic materials. *Structural Optimization*, 1 (2):101–106, 1989. ISSN 09344373. doi: 10.1007/BF01637666.

[28] P. Pedersen. On thickness and orientational design with orthotropic materials. *Structural Optimization*, 3(2):69–78, 1991. ISSN 09344373. doi: 10.1007/BF01743275.

[29] J. Thomsen. Optimization of composite discs. *Structural Optimization*, 3(2):89–98, 1991. ISSN 09344373. doi: 10.1007/BF01743277.

[30] Albert A. Groenwold and Raphael T. Haftka. Optimization with non-homogeneous failure criteria like Tsai-Wu for composite laminates. *Structural and Multidisciplinary Optimization*, 32(3):183–190, 2006. ISSN 1615147X. doi: 10.1007/s00158-006-0020-3.

[31] M Bruyneel. A general and effective approach for the optimal design of fiber reinforced composite structures. 66:1303–1314, 2006. doi: 10.1016/j.compscitech.2005.10.011.

[32] M Bruyneel. Optimization of laminated composite structures: problems, solution procedures and applications. *Composite Materials Research Progess*, 2008, 2008. URL www.novapublishers.com.

[33] Shun Fa Hwang, Ya Chu Hsu, and Yuder Chen. A genetic algorithm for the optimization of fiber angles in composite laminates. *Journal of Mechanical Science and Technology*, 28(8):3163–3169, 2014. ISSN 1738494X. doi: 10.1007/s12206-014-0725-y.

[34] Kun Li, Shi Lin Yan, Wen Feng Pan, and Gang Zhao. Optimization of fiber-orientation distribution in fiber-reinforced composite injection molding by Taguchi, back propagation neural network, and genetic algorithm-particle swarm optimization. *Advances in Mechanical Engineering*, 9(9):1–11, 2017. ISSN 16878140. doi: 10.1177/1687814017719221.

[35] Camila Aparecida Diniz, Sebastião Simões Cunha, Guilherme Ferreira Gomes, and Antônio Carlos Ancelotti. Optimization of the Layers of Composite Materials from Neural Networks with Tsai–Wu Failure Criterion. *Journal of Failure Analysis and Prevention*, 19(3):709–715, 2019. ISSN 15477029. doi: 10.1007/s11668-019-00650-w.

[36] Andre Luis Ferreira da Silva, Ruben Andres Salas, Emilio Carlos Nelli Silva, and J. N. Reddy. Topology optimization of fibers orientation in hyperelastic composite material. *Composite Structures*, 231(February 2019):111488, 2020. ISSN 02638223. doi: 10.1016/j.compstruct.2019.111488. URL https://doi.org/10.1016/j.compstruct.2019.111488.

[37] Yang Shen and David Branscomb. Orientation optimization in anisotropic materials using gradient descent method. *Composite Structures*, 234(September 2019):111680, 2020. ISSN 02638223. doi: 10.1016/j.compstruct.2019.111680. URL https://doi.org/10.1016/j.compstruct.2019.111680.

[38] Gianluigi Rozza and Karen Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1244–1260, 2007. ISSN 00457825. doi: 10.1016/j.cma.2006.09.005.

[39] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1–47, 2007. ISSN 1134-3060. doi: 10.1007/bf03024948.

[40] G. Rozza, D. B.P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008. ISSN 11343060. doi: 10.1007/s11831-008-9019-9.

[41] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: An introduction*. 2015. ISBN 9783319154312. doi: 10.1007/978-3-319-15431-2.

[42] Y. C. Liang, H. P. Lee, S. P. Lim, W. Z. Lin, K. H. Lee, and C. G. Wu. Proper orthogonal decomposition and its applications - Part I: Theory. *Journal of Sound and Vibration*, 252(3):527–544, 2002. ISSN 0022460X. doi: 10.1006/jsvi.2001.4041.

[43] Claudio Maccone. *Deep Space Flight and Communications*. 2009. ISBN 9783540729426. doi: 10.1007/978-3-540-72943-3.

[44] C. Prud'homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering, Transactions of the ASME*, 124(1):70–80, 2002. ISSN 00982202. doi: 10.1115/1.1448332.

[45] Gianluigi Rozza, D. B.P. Huynh, and Andrea Manzoni. Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants. 2013.

[46] A. Ammar, F. Chinesta, P. Diez, and A. Huerta. An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering*, 199(25-28):1872–1880, 2010. ISSN 00457825. doi: 10.1016/j.cma.2010.02.012.

[47] F. Chinesta and E. Cueto. *PGD-Based Modeling of Materials, Structures and Processes*. 2014. ISBN 978-3-319-06181-8. doi: 10.1007/978-3-319-06182-5.

[48] Francisco Chinesta and Pierre Ladevèze. *Separated Representations and PGD-Based Model Reduction*, volume 554. 2014. ISBN 978-3-7091-1793-4. doi: 10.1007/978-3-7091-1794-1. URL http://link.springer.com/10.1007/978-3-7091-1794-1.

[49] Francisco Chinesta, Adrien Leygue, Felipe Bordeu, Elias Cueto, David Gonzalez, Amine Ammar, and Antonio Huerta. PGD-Based Computational Vademecum for Efficient Design , Optimization and Control. *Archives of Computa- tional Methods in Engineering*, 20(1):31–59, 2017. doi: 10.1007/s11831-013-9080-x.

[50] Pedro Díez, Sergio Zlotnik, Alberto García-González, and Antonio Huerta. Algebraic PGD for tensor separation and compression: An algorithmic approach. *Comptes Rendus - Mecanique*, 346(7):501–514, 2018. ISSN 16310721. doi: 10.1016/j.crme.2018.04.011.

[51] P. Díez, S. Zlotnik, A. García-González, and A. Huerta. Encapsulated PGD Algebraic Toolbox Operating with High-Dimensional Data. *Archives of Computational Methods in Engineering*, 26(5), 2019. ISSN 18861784. doi: 10.1007/s11831-019-09378-0. URL https://doi.org/10.1007/s11831-019-09378-0.

[52] P. Vidal, L. Gallimard, and O. Polit. Proper Generalized Decomposition and layer-wise approach for the modeling of composite plate structures. *International Journal of Solids and Structures*, 50(14-15):2239–2250, 2013. ISSN 00207683. doi: 10.1016/j.ijsolstr.2013.03.034. URL http://dx.doi.org/10.1016/j.ijsolstr.2013.03.034.

[53] P. Vidal, L. Gallimard, and O. Polit. Shell finite element based on the Proper Generalized Decomposition for the modeling of cylindrical composite structures. *Computers and Structures*, 132:1–11, 2014. ISSN 00457949. doi: 10.1016/j.compstruc.2013.10.015. URL http://dx.doi.org/10.1016/j.compstruc.2013.10.015.

[54] A. Leygue and E. Verron. A First Step Towards the Use of Proper General Decomposition Method for Structural Optimization. *Archives of Computational Methods in Engineering*, 17(4):465–472, 2010. ISSN 11343060. doi: 10.1007/s11831-010-9052-3.

[55] Amine Ammar, Antonio Huerta, Francisco Chinesta, Elías Cueto, and Adrien Leygue. Parametric solutions involving geometry: A step towards efficient shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268:178–193, 2013. ISSN 00457825. doi: 10.1016/j.cma.2013.09.003. URL http://dx.doi.org/10.1016/j.cma.2013.09.003.

[56] Alberto Sibileau, Alberto García-González, Ferdinando Auricchio, Simone Morganti, and Pedro Díez. Explicit parametric solutions of lattice structures with proper generalized decomposition (PGD): Applications to the design of 3D-printed architectured materials. *Computational Mechanics*, 62(4):871–891, 2018. ISSN 01787675. doi: 10.1007/s00466-017-1534-9.

[57] F. Chinesta, A. Ammar, A. Leygue, and R. Keunings. An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11):578–592, 2011. ISSN 03770257. doi: 10.1016/j.jnnfm.2010.12.012.

[58] David Modesto, Sergio Zlotnik, and Antonio Huerta. Proper generalized decomposition for parameterized Helmholtz problems in heterogeneous and unbounded domains: Application to harbor agitation. *Computer Methods in Applied Mechanics and Engineering*, 295:127–149, 2015. ISSN 00457825. doi: 10.1016/j.cma.2015.03.026.

[59] Pedro Díez, Sergio Zlotnik, and Antonio Huerta. Generalized parametric solutions in Stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 326:223–240, 2017. ISSN 00457825. doi: 10.1016/j.cma.2017.07.016.

[60] R. García-Blanco, D. Borzacchiello, F. Chinesta, and P. Diez. Monitoring a PGD solver for parametric power flow problems with goal-oriented error assessment. *International Journal for Numerical Methods in Engineering*, 111(6):529–552, 2017. ISSN 10970207. doi: 10.1002/nme.5470.

[61] Jose V. Aguado, Antonio Huerta, Francisco Chinesta, and Elías Cueto. Real-time monitoring of thermal processes by reduced-order modeling. *International Journal for Numerical Methods in Engineering*, (February):1102–1119, 2014. doi: 10.1002/nme.4784. URL http://onlinelibrary.wiley.com/doi/10.1002/nme.3279/full.

[62] Hasini Garikapati, Sergio Zlotnik, Pedro Díez, Clemens V. Verhoosel, and E. Harald van Brummelen. A Proper Generalized Decomposition (PGD) approach to crack propagation in brittle materials: with application to random field material properties. *Computational Mechanics*, 65(2):451–473, 2020. ISSN 14320924. doi: 10.1007/s00466-019-01778-0. URL https://doi.org/10.1007/s00466-019-01778-0.

[63] Marianna Signorini, Sergio Zlotnik, and Pedro Díez. Proper generalized decomposition solution of the parameterized Helmholtz problem: application to inverse geophysical problems. *International Journal for Numerical Methods in Engineering*, 109(8):1085–1102, 2017. ISSN 10970207. doi: 10.1002/nme.5313.

[64] Julien Berger, Helcio R.B. Orlande, and Nathan Mendes. Proper Generalized Decomposition model reduction in the Bayesian framework for solving inverse heat transfer problems. *Inverse Problems in Science and Engineering*, 25(2):260–278, 2017. ISSN 17415985. doi: 10.1080/17415977.2016.1160395. URL http://dx.doi.org/10.1080/17415977.2016.1160395.

[65] Zachary M. Prince and Jean C. Ragusa. Parametric uncertainty quantification using proper generalized decomposition applied to neutron diffusion. *International Journal for Numerical Methods in Engineering*, 119(9):899–921, 2019. ISSN 10970207. doi: 10.1002/nme.6077.

[66] Amaury Courard, David Néron, Pierre Ladevèze, and Ludovic Ballere. Integration of PGD-virtual charts into an engineering design process. *Computational Mechanics*, 57(4):637–651, 2016. ISSN 01787675. doi: 10.1007/s00466-015-1246-y.

[67] Ted Fish, Jacob, Belytschko. *A first Course in Finite Elements Method*. Wiley, 2007. ISBN 9788578110796. doi: 10.1017/CBO9781107415324.004.

[68] Yakov Itin and Friedrich W. Hehl. The constitutive tensor of linear elasticity: Its decompositions, Cauchy relations, null Lagrangians, and wave propagation. *Journal of Mathematical Physics*, 54(4), 2013. ISSN 00222488. doi: 10.1063/1.4801859.

[69] Robert M. Jones. *Mechanics of composite materials*. Taylor & Francis, 1999. ISBN 1-56032-712-X.

[70] Daniel Ishai, Isaac M. , Ori. *Engineering Mechanics of Composite Materials*. Oxford University Press, second edition, 2006. ISBN 978-0-19-515097-1.

[71] P. K. Mallick. *Fibre-Reinforced Composites*. Taylor & Francis, 2007.

[72] Jack R Vinson and Robert L Sierakowski. The Behavior of Structures Composed of Composite Materials Solid Mechanics and its Applications Volume 105 Series Editor. *Civil Engineering*, 105:445, 2002. URL http://books.google.com/books?hl=en{&}lr={&}id=dgUl3Grle1MC{&}oi=fnd{&}pg=PR7{&}dq=The+Behavior+of+Structures+Composed+of+Composite+Materials{&}ots=-0VLCP3jdl{&}sig=FE9ik4bIYSI8pefyh6N5R8T08kQ.

[73] Stephen W. Tsai and Edward M. Wu. A General Theory of Strength for Anisotropic Materials. *Journal of Composite Materials*, 5(1):58–80, 1971. ISSN 1530793x. doi: 10.1177/002199837100500106.

[74] Xin-She Yang. *Engineering Optimization*, volume 27. 1976. ISBN 9780470582466. doi: 10.2307/3009231.

[75] Sergio Zlotnik, Pedro Díez, David Modesto, and Antonio Huerta. Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *International Journal for Numerical Methods in Engineering*, 103(10):737–758, 2015. ISSN 10970207. doi: 10.1002/nme.4909.

[76] X. Zou, M. Conti, P. Díez, and F. Auricchio. A nonintrusive proper generalized decomposition scheme with application in biomechanics. *International Journal for Numerical Methods in Engineering*, 113(2):230–251, 2018. ISSN 10970207. doi: 10.1002/nme.5610.

[77] Amine Ammar, Francisco Chinesta, Elías Cueto, Manuel Doblare, Amine Ammar, Francisco Chinesta, Elías Cueto, Manuel Doblare, and Proper Generalized. Proper Generalized Decomposition of Time-Multiscale Models. *International Journal for Numerical Methods in Engineering*, 83:0–28, 2016.

[78] Alexander Kißling, Frank Beneke, and Thomas Seul. Determination of anisotropic material properties of carbon-fiber-reinforced FDM structures for numerical simulations. *Annual Technical Conference - ANTEC, Conference Proceedings*, 2017-May:27–34, 2017.

[79] A. Falcó and A. Nouy. A Proper Generalized Decomposition for the solution of elliptic problems in abstract form by using a functional Eckart-Young approach. *Journal of Mathematical Analysis and Applications*, 376(2):469–480, 2011. ISSN 0022247X. doi: 10.1016/j.jmaa.2010.12.003. URL http://dx.doi.org/10.1016/j.jmaa.2010.12.003.

[80] Ludovic Chamoin, Florent Pled, Pierre Eric Allier, and Pierre Ladevèze. A posteriori error estimation and adaptive strategy for PGD model reduction applied to parametrized linear parabolic problems. *Computer Methods in Applied Mechanics and Engineering*, 327:118–146, 2017. ISSN 00457825. doi: 10.1016/j.cma.2017.08.047.

[81] Keh Shih Chuang, Hong Long Tzeng, Sharon Chen, Jay Wu, and Tzong Jer Chen. Fuzzy c-means clustering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics*, 30(1):9–15, 2006. ISSN 08956111. doi: 10.1016/j.compmedimag.2005.10.001.

[82] Pall F. Jonsson, Tamara Cavanna, Daniel Zicha, and Paul A. Bates. Cluster analysis of networks generated through homology: Automatic identification of important protein communities involved in cancer metastasis. *BMC Bioinformatics*, 7:1–13, 2006. ISSN 14712105. doi: 10.1186/1471-2105-7-2.

[83] Author M Claire Horner-devine and Brendan J M Bohannan. Phylogenetic Clustering and Overdispersion in Bacterial Communities Published by : Ecological Society of America PHYLOGENETIC CLUSTERING AND OVERDISPERSION IN BACTERIAL COMMUNITIES. *Wiley*, 87(7), 2016.

[84] Girish Punj and David W. Stewart. Cluster Analysis in Marketing Research: Review and Suggestions for Application. *Journal of Marketing Research*, 20(2):134, 1983. ISSN 00222437. doi: 10.2307/3151680.

[85] Radan Huth, Christoph Beck, Andreas Philipp, Matthias Demuzere, Zbigniew Ustrnul, Monika Cahynová, Jan Kyselý, and Ole Einar Tveito. Classifications of atmospheric circulation patterns: Recent advances and applications. *Annals of the New York Academy of Sciences*, 1146:105–152, 2008. ISSN 17496632. doi: 10.1196/annals.1446.019.

[86] Olfa Nasraoui and Chiheb-Eddine Ben N'Cir. *Clustering Methods for Big Data Analytics*. Springer, 2019. ISBN 978-3-319-97863-5. doi: 10.1007/978-3-319-97864-2. URL https://www.springer.com/gp/book/9783319978635{%}0Ahttp://link.springer.com/10.1007/978-3-319-97864-2.

[87] Volodymyr Melnykov and Xuwen Zhu. An extension of the K-means algorithm to clustering skewed data. *Computational Statistics*, 34(1):373–394, 2019. ISSN 16139658. doi: 10.1007/s00180-018-0821-z. URL https://doi.org/10.1007/s00180-018-0821-z.

[88] D. J. Bartholomew. Principal components analysis. *International Encyclopedia of Education*, 2010. doi: 10.1016/B978-0-08-044894-7.01358-0.

[89] Cosma Rohilla Shalizi. Advanced data analysis from an elementary point of view. *Book Manuscript*, page 801, 2013. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.371.4613{&}rep=rep1{&}type=pdf.

[90] Mostafa Akbari, Parviz Asadi, Mohammad Kazem Besharati Givi, and Ghazaleh Khodabandehlouie. *Artificial neural network and optimization*. Elsevier Ltd, 2014. ISBN 9780857094551. doi: 10.1533/9780857094551.543.

[91] Mark Cotteleer, Monica Mahto, and Jonathan Holdowsky. The 3D opportunity primer. *A Deloitte series on additive manufacturing*, pages 2 – 14, 2013. URL http://dupress.com/collection/3d-opportunity.

[92] I Gibson, D Rosen, and B Stucker. *Additive Manufacturing Technologies*. Springer, second edition, 2015. ISBN 978-1-4939-2113-3. doi: 10.1007/978-1-4939-2113-3.

[93] Pedram Parandoush and Dong Lin. A review on additive manufacturing of polymer-fiber composites. *Composite Structures*, 182:36–53, 2017. ISSN 02638223. doi: 10.1016/j.compstruct.2017.08.088. URL https://doi.org/10.1016/j.compstruct.2017.08.088.

[94] Sadben Khan, Kazem Fayazbakhsh, Zouheir Fawaz, and Mahdi Arian Nik. Curvilinear variable stiffness 3D printing technology for improved open-hole tensile strength. *Additive Manufacturing*, 24:378–385, 2018. ISSN 22148604. doi: 10.1016/j.addma.2018.10.013. URL https://doi.org/10.1016/j.addma.2018.10.013.

[95] Gianluca Alaimo, Stefania Marconi, Luca Costato, and Ferdinando Auricchio. Influence of meso-structure and chemical composition on FDM 3D-printed parts. *Composites Part B: Engineering*, 113:371–380, 2017. ISSN 13598368. doi: 10.1016/j.compositesb.2017.01.019. URL http://dx.doi.org/10.1016/j.compositesb.2017.01.019.

[96] Mohammad T. Hayasi and Bahram Asiabanpour. A new adaptive slicing approach for the fully dense freeform fabrication (FDFF) process. *Journal of Intelligent Manufacturing*, 24(4):683–694, 2013. ISSN 09565515. doi: 10.1007/s10845-011-0615-4.

[97] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. ISSN 01678655. doi: 10.1016/j.patrec.2009.09.011. URL http://dx.doi.org/10.1016/j.patrec.2009.09.011.

[98] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 07-09-Janu: 1027–1035, 2007.

[99] Renato Cordeiro De amorim. Feature Relevance in Ward's Hierarchical Clustering Using the Lp Norm. *Journal of Classification*, 33:52–72, 2015. doi: 10.1007/s00357.

[100] Frank Nielsen. *Introduction to HPC with MPI for Data Science*. Number February 2016. Springer, 2016. ISBN 978-3-319-21903-5. doi: 10.1007/978-3-319-21903-5. URL http://www.springer.com/series/7592.

[101] Brian Everitt. *Cluster analysis*. Number 1. Wiley, 5 edition, 2011. ISBN 978-0-470-74991-3. doi: 10.1007/BF00154794.

[102] Gabor J Szekely and Rizzo M. L. Hierarchical Clustering_Via_Joint_Between_Within.Pdf, 2005.

[103] Fionn Murtagh and Pierre Legendre. Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion? *Journal of Classification*, 33(October):52–72, 2014. doi: 10.1007/s00357-014-9161-z.